

UNIVERSIDAD DE CUENCA



FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

“IMPLEMENTACIÓN DE LOS PROCESOS RELACIONADOS A LA CAPA
FÍSICA DE UNA RED LTE UTILIZANDO MATLAB”

Tesis previa a la obtención del título de
Ingeniero en Electrónica y Telecomunicaciones

Autores:

Paúl Esteban Cárdenas Delgado.

CI: 010520098-4

Josué Bernardo Vanegas Samaniego.

CI: 010446692-5

Director:

Ing. Édgar Efraín Ochoa Figueroa.

CI: 010213457-4

Cuenca - Ecuador

Marzo-2017

Resumen

En el presente proyecto se realiza un estudio de la capa física de la tecnología 4G LTE, el cual abarca los procesos de codificación, detección y corrección de errores, modulación, multiplexación de frecuencias (OFDM), y las técnicas MIMO para la transmisión de la información, basándose en la normativa 3GPP. A partir de este estudio, se desarrolla una herramienta en el entorno de Matlab, enfocada para docentes y estudiantes, que permite la modelación de cada uno de los procesos que intervienen en la capa física, en la cual se puede observar paso a paso cada uno de ellos.

Esta herramienta permite replicar los procesos del enlace ascendente como del enlace descendente de forma individual, así como la ejecución automática del sistema; dentro de estas opciones se puede observar mediciones de CQI, BER, verificar procesos de modulación, asignación de recursos, construcción de la rejilla OFDMA y SC-FDMA, técnicas MIMO, entre otras.

Palabras clave: Long Term Evolution (LTE), Capa Física, Codificación, Modulación, Enlace Ascendente LTE, Enlace Descendente LTE, Multiplexación de Frecuencias, OFDM, Técnicas MIMO.

Paúl Cárdenas D.
Josué Vanegas S.

Abstract

The aim in this project is to study the physical layer of the 4G LTE technology, which comprises the following processes: encoding, detection and correction of errors, modulation, orthogonal frequency-division multiplexing (OFDM), and MIMO techniques for the information transmission, based on the 3GPP standard. From this study, a tool is developed in the Matlab environment, to support teachers and students. The developed software allows the modeling of each of the processes involved in the physical layer, and you can observe step by step each one of them.

This tool allows to replicate the uplink and downlink processes individually, as well as the automated execution of the system. Within these options we can observe measurements of CQI, BER, verify modulation processes, resource allocation, OFDMA and SC-FDMA grid construction, MIMO techniques, among others.

Keywords: Long Term Evolution (LTE), Physical Layer, Encoding, Modulation, LTE Uplink, LTE Downlink, Orthogonal Frequency-Division Multiplexing OFDM, MIMO Techniques.

Paúl Cárdenas D.
Josué Vanegas S.

Índice general

Resumen	2
Abstract	4
Índice general	4
Índice de figuras.	7
Índice de tablas.	11
1. Introducción	19
1.1. Objetivo General.	20
1.2. Trabajos Relacionados	20
2. Red 4G LTE.	23
2.1. Introducción	23
2.2. Arquitectura de la red de Comunicaciones 4G LTE	25
2.2.1. Núcleo de Paquete EPC	26
2.2.2. Interfaz Aire	27
2.3. Capa Física Red 4G LTE	28
2.3.1. Procesamiento del Canal de Enlace Descendente	32
2.3.2. Procesamiento del Canal de Enlace Ascendente	55
2.3.3. Planificación de usuarios	61
2.3.4. Codificación	65
2.3.5. Multiplexación por División de Frecuencias	72
2.3.6. Modulación	75
2.3.7. Modulación QPSK - Modulación de Fase en Cuadratura	75
2.3.8. Modulación QAM- Modulación de Amplitud en Cuadratura	77
2.3.9. Técnicas Multi-Antena (MIMO)	78

3. Procesos de la Capa Física según la 3GPP	81
3.1. Codificación de Canal, Multiplexación y Entrelazado	81
3.1.1. Código de bloque CRC	81
3.1.2. Código de segmentación en bloques y acoplamiento CRC	83
3.1.3. Codificación de Canal LTE	84
3.1.4. Código Convolutacional Tail Biting	84
3.1.5. Turbo Código.	85
3.2. Adaptación de Velocidad	89
3.2.1. Sub-Bloque Entrelazador	89
3.2.2. Buffer Circular y Selección de bits	92
3.3. Aleatorización	94
3.4. Modulaciones	96
3.5. Procesos del Enlace Descendente	99
3.6. Procesos del Enlace Ascendente	100
3.7. Multiplexación de Frecuencias	100
3.7.1. Estructura del Slot	101
4. Ejemplo Ilustrativo	106
4.1. Introducción	106
4.2. Canal de Codificación	106
4.2.1. Codificación CRC	107
4.2.2. Segmentación de código de bloque	109
4.2.3. Turbo Código	109
4.3. Adaptación de Velocidad	114
4.4. Aleatorización (<i>Scrambling</i>)	119
4.5. Modulaciones	121
4.6. Multiplexación de Frecuencias (OFDM)	121
4.7. MIMO	124
4.8. Receptor	125
5. Descripción del Software	130
5.1. Introducción	130
5.2. Estructura del software	131
5.3. Desarrollo e implementación de algoritmos	135
5.4. Validación del software	137
5.4.1. Codificación CRC	137



5.4.2.	Segmentación de bloque de código	138
5.4.3.	Turbo código	139
5.4.4.	Adaptación de tasa	142
5.4.5.	Aleatorización	144
5.4.6.	Modulación	146
5.4.7.	Demodulación	148
5.4.8.	Des-aleatorización	149
5.4.9.	Decodificación	150
5.4.10.	Codificación CRC	151
5.5.	Simulaciones realizadas	152
5.5.1.	Primera simulación	154
5.5.2.	Segunda simulación	159
5.5.3.	Tercera simulación	164
6.	Conclusiones y Recomendaciones.	167
6.1.	Conclusiones	167
6.2.	Recomendaciones	171
A.	Manual de Usuario	173
A.1.	Menú Inicial	173
A.2.	Estructura de ventanas	174
A.2.1.	Ventana de envío	174
A.2.2.	Ventana de Recepción	176
A.3.	Enlace Ascendente	177
A.4.	Enlace Descendente	189
A.5.	Procesos	193
A.5.1.	Procesos de envío	193
A.5.2.	Procesos recepción	204
B.	Glosario de términos	210
C.	Oficio ARCOTEL	213

Índice de figuras.

2.1. Arquitectura de la red 4G LTE. [3]	26
2.2. Arquitectura de un EPC de la red 4G LTE. [3]	27
2.3. Protocolos de la interfaz aire en LTE. [9]	28
2.4. Procesos de la capa física LTE para el enlace descen- dente. [16]	31
2.5. Procesos de la Capa física LTE para el enlace ascen- dente. [16]	33
2.6. Procesos de los canales físicos enlace descendente. [2]	34
2.7. Canales del enlace descendente.[9]	35
2.8. Procesamiento hacia la codificación del canal. [2] . . .	36
2.9. Codificación del canal de enlace descendente. [2] . . .	37
2.10. Aleatorización (<i>Scrambling</i>) y modulación. [2]	38
2.11. Modulación OFDM. [2]	41
2.12. Generación del canal PDSCH. [2]	43
2.13. Mapeo del canal PBCH. [2]	45
2.14. Generación del canal PCFICH. [2]	47
2.15. Solicitud de repetición automática híbrida. [13]	49
2.16. Generación del canal PHICH. [2]	50
2.17. Posición de los canales físicos del enlace descendente. [6]	51
2.18. Tiempo y Frecuencia de los canales físicos enlace des- cendente. [6]	51
2.19. Posición señales físicas PSS y SSS. [2]	53
2.20. Posición señales físicas configuración básica. [6] . . .	54
2.21. Canales del enlace ascendente. [9]	55
2.22. Canal físico PUSCH. [6]	56
2.23. Canal físico PUCCH. [6]	56

2.24. Canal físico PRACH. [6]	57
2.25. Señal de referencia DMRS. [2]	58
2.26. Procesamiento UL-SCH. [2]	59
2.27. Procesamiento PUSCH. [2]	59
2.28. Multiplexación de datos y control. [2]	60
2.29. Diagrama de procesamiento PUSCH. [2]	61
2.30. BLER vs relación señal/ruido (SNR) para diferentes MCS. [14]	63
2.31. Codificador convolucional $k = 1, n = 3, m = 3$ y tasa $R = 1/3$. [3]	66
2.32. Estructura de codificador turbo para LTE. [3]	67
2.33. Diagrama de Trellis. [4]	70
2.34. Representación de una señal OFDM. [3]	73
2.35. Asignación de en tiempo y frecuencia OFDMA. [3] . .	74
2.36. Diferencias entre subportadoras. [2]	74
2.37. Constelación de modulación QPSK. [5]	76
2.38. Constelación de modulación 16-QAM. [5]	77
2.39. Técnica MIMO configuración 2x2. [3]	78
3.1. Codificador convolucional con $k = 1, n = 3, m = 6$ tasa $R = 1/3$. [1]	85
3.2. Estructura de codificador turbo con tasa 1/3. [3] . . .	85
3.3. Procesos del <i>rate matching</i> . [1]	89
3.4. Diagrama secuencias gold. [4]	96
3.5. Pasos para la codificación del canal BCH. [1]	99
3.6. Procesamiento del bloque de transporte para DL- SCH, PCH y MCH. [1]	99
3.7. Procesamiento del Bloque de transporte para UL- SCH. [1]	100
3.8. Estructura de la trama correspondiente a FDD. [1] .	101
3.9. Estructura del slot para el enlace ascendente. [1] . .	102
3.10. Estructura del <i>slot</i> para el enlace descendente. [1] . .	104
3.11. Prefijo cíclico evita ISI. [1]	105
4.1. Codificador Turbo. [3]	107
4.2. Rejilla de Recursos para una Subtrama.	123
4.3. Técnicas Multiantenas. [16]	124



4.4. Mapeo de los símbolos.	125
4.5. Diagrama de Trellis para la salida parcial.	128
5.1. Diagrama de bloques del software.	131
5.2. Procesos envío y recepción de datos.	131
5.3. Procesos bloques de código.	132
5.4. Procesos palabra de código.	132
5.5. Codificación CRC, software implementado.	137
5.6. Codificación CRC, MATLAB.	138
5.7. Segmentación de bloque de código.	138
5.8. Turbo código, software implementado.	140
5.9. Primer codificador convolucional, MATLAB.	141
5.10. Segundo codificador convolucional, MATLAB.	141
5.11. Adaptación de tasa, software implementado.	143
5.12. Adaptación de tasa, MATLAB.	144
5.13. Aleatorización, software implementado.	145
5.14. Aleatorización, MATLAB.	146
5.15. Modulación, software.	147
5.16. Símbolos, software.	147
5.17. Modulación y símbolos, MATLAB.	147
5.18. Demodulación, software.	148
5.19. Demodulación, MATLAB.	148
5.20. Des-aleatorización, software.	149
5.21. Des-aleatorización, MATLAB.	150
5.22. Decodificación, software.	151
5.23. Decodificación, MATLAB.	151
5.24. Proceso CRC recepción, software.	152
5.25. Proceso CRC recepción, MATLAB.	152
5.26. Capas MIMO.	154
5.27. Rejilla OFDM.	155
5.28. Espectro de frecuencia.	155
5.29. Procesos y subportadoras.	156
5.30. Datos de salida.	156
5.31. Datos MIMO/OFDM.	157
5.32. Subportadoras recibidas con ruido, BER=0.	157
5.33. Diagrama de constelación.	157
5.34. Diagrama de Trellis.	158

5.35. Datos recibidos, BER=0, CQI, ACK.	158
5.36. Capas MIMO.	159
5.37. Rejilla OFDM.	160
5.38. Diagrama de constelación recibido.	160
5.39. BER información recibido.	161
5.40. Diagrama de Trellis.	161
5.41. BER con corrección de errores.	161
5.42. Diagrama de constelación.	162
5.43. BER información recibido.	162
5.44. Diagrama de Trellis.	163
5.45. BER con corrección de errores.	163
5.46. Modulación 64QAM.	164
5.47. BER con corrección de errores.	165
5.48. Modulación 16QAM.	165
5.49. BER con corrección de errores.	165
5.50. Modulación QPSK.	166
5.51. BER con corrección de errores.	166

desde 1867

Índice de tablas.

2.1. Mapeo de capas multiplexación espacial. [2]	39
2.2. Mapeo de capas diversidad de transmisión. [2]	40
2.3. Número de símbolos de control. [2]	46
2.4. Formato PDCCH. [2]	50
2.5. Valores de CQI. [14]	64
2.6. Estados de Trellis. [16]	68
3.1. Esquema y velocidad de codificación para el canal de transporte. [1]	84
3.2. Esquema y velocidad de codificación para el canal de control. [1]	84
3.3. Valores de f_1 y f_2 de acuerdo al tamaño de bloque K. [1]	88
3.4. Modulación BPSK. [1]	96
3.5. Modulación QPSK. [1]	97
3.6. Modulación 16-QAM. [1]	97
3.7. Modulación 64-QAM. [1]	98
3.8. Parámetros del bloque de recurso enlace ascendente. [1]	103
3.9. Parámetros del bloque de recurso enlace descendente. [1]	103
4.1. Codificación letra H.	106
4.2. Codificador convolucional I.	110
4.3. Entrelazador (<i>Interleaver</i>).	112
4.4. Codificador convolucional II.	113



Paúl Esteban Cárdenas Delgado, autor/a del Trabajo de Titulación "Implementación de los procesos relacionados a la capa física de una red LTE utilizando Matlab", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniero en Electrónica y Telecomunicaciones. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor/a

Cuenca, 09 de Mayo de 2017

Paúl Esteban Cárdenas Delgado

C.I: 010520098-4



Josué Bernardo Vanegas Samaniego, autor/a del Trabajo de Titulación "Implementación de los procesos relacionados a la capa física de una red LTE utilizando Matlab", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniero en Electrónica y Telecomunicaciones. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor/a

Cuenca, 09 de Mayo de 2017

Josué Bernardo Vanegas Samaniego

C.I: 010446692-5



Paúl Esteban Cárdenas Delgado, autor del Trabajo de Titulación "Implementación de los procesos relacionados a la capa física de una red LTE utilizando Matlab", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a.

Cuenca, 09 de Mayo de 2017

Paúl Esteban Cárdenas Delgado

C.I: 010520098-4



Josué Bernardo Vanegas Samaniego, autor del Trabajo de Titulación "Implementación de los procesos relacionados a la capa física de una red LTE utilizando Matlab", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a.

Cuenca, 09 de Mayo de 2017

Josué Bernardo Vanegas Samaniego

C.I: 010446692-5

Agradecimientos

Una tesis de grado no requiere solamente del esfuerzo personal de quien la realiza, sino del apoyo tanto en lo académico como en lo personal de muchas personas, escribo estas líneas en agradecimiento a quienes me han brindado su ayuda y apoyo, no solo durante el desarrollo de la tesis, sino durante toda mi carrera.

A mis padres, Anita y Diego, y hermanos, Juan y Pablo, por todo el cariño y apoyo que me han dado y por estar siempre a mi lado, no solo ahora, sino siempre.

A mi novia María Belén, por el amor, paciencia y compañía, que me dio en cada momento, y sobre todo por haber estado conmigo apoyándome, y por haberme motivado siempre a seguir mis sueños y nunca rendirme.

A mi compañero de tesis, quien trabajó arduamente conmigo para conseguir este objetivo.

A mis directores, Ingeniera Sofía Arévalo y el Ingeniero Edgar Ochoa, por su tiempo, ayuda, consejos, y gran colaboración en cada momento de consulta para el desarrollo de este trabajo.

Al ingeniero Esteban Andrade, quien nos brindó bibliografía muy importante para nuestro proyecto.

A todos los profesores que me han brindado sus conocimientos durante todos estos años.

A todos mis amigos que me apoyaron durante estos años, a todos, mi más grande agradecimiento.

Paúl Cárdenas D.

Agradecimientos

Concluir este propósito académico requiere un gran empeño y dedicación, no solo académico sino también de personas que día a día contribuyeron en este largo camino, es por esto que deseo expresarles mi sincero agradecimiento.

Primero quiero agradecer a Dios por haberme dado la sabiduría y la inteligencia para poder culminar esta etapa.

A mis padres y hermanos, quienes estuvieron apoyándome en todas las etapas de esta carrera universitaria, con sus constantes palabras de aliento.

A los ingenieros Édgar Ochoa y Sofía Arévalo, por haber aceptado dirigir esta tesis, por su esfuerzo, ayuda, y tiempo empleado; no solo en el desarrollo de este proyecto sino también por los conocimientos brindados en las aulas de clases.

A mi compañero de tesis Paúl, por su esfuerzo y dedicación para poder culminar este proyecto.

A Daniela, que me acompañó a lo largo de la carrera universitaria, brindándome su apoyo en cada momento, gracias por su compañía y colaboración en esta etapa final.

A mis profesores, quienes impartieron sus conocimientos, colaborando de esta manera a mi formación profesional.

A mis amigos, Iván, Gonzalo, Gino, Andrés, Xavier, David, Martín, quienes día a día me apoyaron, a todos ustedes mi agradecimiento.

Josué Vanegas S.

Dedicatoria

Este trabajo lo quiero dedicar en las siguientes líneas a personas a las que quiero mucho, que son indispensables en mi vida y que han estado a mi lado y me han dado mucho más que su apoyo.

Dedico este trabajo, y sobre todo el título que me permitirá conseguir, fruto de un gran esfuerzo, sacrificio y dedicación, a una persona que llevo siempre en mi mente y en mi corazón, que ahora no está presente, pero que me dejó muchísimas enseñanzas, y quien fue un ejemplo de fortaleza, lucha y perseverancia, quien me enseñó y siempre me motivo al estudio, a siempre querer aprender más, a nunca rendirme y sobre todo a seguir adelante a pesar cualquier adversidad, alguien que siempre me dio su cariño y que sé que desde algún lugar está pendiente de cada paso en mi vida, a mi muy querido y recordado abuelito Néstor Delgado Jaramillo, quien es mi motivación para seguir adelante y luchar por conseguir mis sueños.

También se lo dedico a mis padres, porque representan un pilar fundamental en mi vida, porque con su ejemplo, consejos y sobre todo su amor, me han hecho la persona que soy, y estoy seguro que sin ellos sería imposible cumplir esta meta.

A mis hermanos y a mis sobrinos Juliana y Juan Martin, que son una alegría inmensa en mi vida, y sin duda una motivación para seguir adelante cada día.

Finalmente a una persona muy importante para mí, que se ha convertido en la mejor compañera de mi vida, y que me llena de felicidad cada día, mi novia María Belén, quien desde que está a mi lado cambio mi vida, quien me ha dado tanto amor y cariño, y ha sido mi motivación para cumplir esta, y muchas metas más.

Paúl Cárdenas D.

Dedicatoria

Este trabajo quiero dedicar a mis padres, Patricio y Lilian, que gracias a su esfuerzo y palabras, no permitieron que me rinda en este camino, gracias por brindarme sus consejos y valores, y sobre todo por enseñarme que este título es una bendición que Dios me ha regalado.

También, quiero dedicar a mis abuelitos, Leopoldo y Baldomira, Germán y Mercedes; de quienes tuve un apoyo incondicional.

A mis hermanos, David, Paz, Juan e Isaac, que son mi motivación para seguir adelante.

Josué Vanegas S.

Capítulo 1

Introducción

En el área de las telecomunicaciones, se han ido desarrollando a lo largo de los años varias tecnologías móviles, las mismas que han respondido correctamente a las necesidades y demandas de los usuarios en el momento adecuado, y continuamente han evolucionado de acuerdo a las tecnologías y servicios que se brindan.

La tecnología para sistemas de comunicación móviles 4G LTE es la más utilizada actualmente por las operadoras de telefonía móvil a nivel mundial [3], en Ecuador no es la excepción, dentro de la ciudad de Cuenca las operadoras móviles disponen de esta tecnología, en donde cada día crece la cobertura dentro de la ciudad.

Si bien esta tecnología comprende el procesamiento de varias capas, en donde cada una de ellas cumple una función específica para lograr la comunicación. La más importante es la capa física y en ésta se realizan los procesos fundamentales que caracterizan a la tecnología LTE.

Actualmente en la Universidad de Cuenca, en la carrera de Ingeniería Electrónica y Telecomunicaciones, se estudia la tecnología LTE como un modelo de comunicación móvil, siendo un tema de alto interés para los estudiantes. Se ha identificado la necesidad de presentar este proyecto de una forma didáctica, ya que actualmente la universidad no posee módulos en donde se pueda simular este tipo de tecnología, por lo que el presente trabajo plantea el desarrollo de una herramienta que proporcione un ambiente de simulación y aprendizaje adecuado de la capa física de LTE, y ayude a comprender de manera más clara los procesos que abarca la misma, y de esta manera complementar los conocimientos teóricos.



1.1. Objetivo General.

Diseñar e implementar los procesos de la capa física de la red LTE (*Long Term Evolution*) empleando el Software con un entorno interactivo Matlab.

Objetivos específicos.

1. Estudiar los procesos que abarca la capa física de la norma LTE según la normativa 3GPP.
2. Estudiar las técnicas de la capa física de la norma LTE.
3. Implementar las técnicas diseñadas en el software Matlab y obtener resultados y conclusiones de las mismas.

1.2. Trabajos Relacionados

En cuánto a simuladores o herramientas de aprendizaje para la tecnología LTE, se han encontrado los siguientes proyectos:

En primer lugar el paper “Plataforma para la Enseñanza de la Capa Física de LTE R8 Usando Matlab” [10] el mismo que fue realizado en noviembre del 2015, en este trabajo se desarrolla una plataforma para la enseñanza de conceptos fundamentales de la capa física de una red LTE Release 8 (R8), utilizando el entorno de programación de MATLAB/SIMULINK. En el desarrollo implementaron el código necesario para obtener como resultados gráficas del BER correspondientes a diferentes modelos de canal y opciones de modulaciones, codificación de canal, ecualización, MIMO (*Multiple Input Multiple Output*), así como observar mallas de recursos y el espectro de señales.

La plataforma dispone de una interfaz gráfica que permite escoger el tipo de simulación que se quiera realizar entre las cuales se encuentran los siguientes módulos:

- Codificación de Banda Base.
- Modulación.



- Modelamiento de Canal.
- SISO (*Single Input, Single Output*).
- SIMO (*Single Input, Multiple Output*).
- Modos de Transmisión.
- Adaptación de Enlace.

Como resultados los autores indican que se logró implementar una plataforma la cual contiene simulaciones de los aspectos fundamentales de la capa física de LTE mediante MATLAB.

Un segundo paper que se revisó, es “Estudio y simulación de turbo códigos utilizando el algoritmo MAP y SOVA” [15]; el cual presenta un análisis teórico de los sistemas de comunicación digital, el canal de comunicaciones, la teoría de codificación e información y los principales teoremas de Shannon, el desarrollo de plataformas de simulación, esquemas de detección y corrección de errores, Turbo Códigos, etc. Para el desarrollo de la herramienta computacional utilizan la plataforma de programación MATLAB. En este paper se logró implementar el simulador de los esquemas de detección de errores, y se dejan algunas recomendaciones importantes después de un análisis de los resultados.

Entre las recomendaciones más importantes, indican que los resultados de las simulaciones han mostrado una gran ventaja en utilizar turbo codificación frente a un canal sin codificación para control de errores, un código de bloque y la codificación convolucional. Otro aspecto importante que se menciona es que en la práctica no se utiliza un número de iteraciones muy alto para el proceso de turbo decodificación, ya que esto implica mayor tiempo de procesamiento por parte de la computadora del receptor, haciendo lento el sistema y ocupando más recursos.

Capítulo 2

Red 4G LTE.

2.1. Introducción

Un sistema de comunicación tiene como objetivo transmitir datos entre dos o más nodos, esperando un alto grado de calidad de servicio (QoS, *Quality of Service*) que es lo que se busca dentro del desarrollo de los diferentes estándares existentes hasta el momento.

El inicio de las tecnologías de comunicación móvil, empezó con el estándar 2G, el cual soporta aplicaciones de telefonía y voz móvil. Posteriormente se desarrolla el estándar 3G el mismo que permite la transmisión de paquetes de datos, gracias a esto se accede a servicios y aplicaciones de internet (correo electrónico, navegación web, mensajería de texto, servicios de cliente-servidor, etc.). Finalmente la demanda de servicios como: video conferencia, voz sobre IP, juegos en línea, *video streaming HD*, exigen el desarrollo de un estándar que permita una velocidad de datos mucho mayor y que no comprometa la calidad de servicio. Para esto se creó el estándar 4G LTE, el cual se caracteriza por transmitir toda la información mediante paquetes IP.

El estándar 4G LTE según las especificaciones en 3GPP (*3rd Generation Partnership Project*) presenta una tasa de transmisión nominal máxima de 1 Gbps para el enlace descendente mientras que para el enlace ascendente son 500 Mbps. Esta transmisión de datos se realiza en la capa física del modelo de comunicación.

LTE está caracterizado principalmente por las técnicas aplicadas en la capa física lo cual garantiza las velocidades ofrecidas y el grado de calidad del servicio que el usuario exige.

La capa física de LTE comprende varios procesos, entre los cuales se destacan la codificación y decodificación, modulación y demodulación, mapeo de capa y el proceso de multiplexación de frecuencias. Las especificaciones técnicas de los procesos de esta tecnología se encuentran establecidos dentro de los estándares 3GPP.

Este proyecto presenta la implementación de un *software* que recopila los aspectos fundamentales de la capa física y los procesos establecidos por el estándar 3GPP.

Esta herramienta se ha desarrollado en MATLAB y está enfocada para estudiantes y docentes; por lo cual presenta una interfaz gráfica amigable para el usuario, la misma que nos permite realizar simulaciones de:

- Enlace ascendente.
- Enlace descendente.
- Simulación automática.

El simulador permite complementar lo aprendido teóricamente y validar paso a paso cada proceso de los canales descendente y ascendente, así como también se puede observar mediciones de CQI, BER, verificar procesos de modulación, asignación de recursos, construcción de la rejilla OFDMA y SC-FDMA, técnicas MIMO, entre otras.

En el Apéndice A, manual de usuario, se puede ver a detalle las funciones y el uso de cada una de las opciones del simulador. El desarrollo de esta herramienta está orientado en dar una explicación paso a paso de cada uno de los procesos que intervienen en la capa física de LTE, por lo que se ha implementado cada uno de los algoritmos utilizados en este *software*, para de esta manera poder mostrar detallada y claramente cada proceso. Para esto no se han utilizado módulos o líneas de comando propias de Matlab, que si bien facilitan el trabajo, pero al no conocer su código, no se puede mostrar los procesos internos de cada una de ellas, dificultando el análisis de dichos procesos.

2.2. Arquitectura de la red de Comunicaciones 4G LTE

La arquitectura de la red 4G LTE del sistema de telecomunicaciones, está conformada por tres componentes, los cuales están representados en la Figura 2.1. Se visualiza que la red que comunica al equipo del usuario (UE¹) con la red troncal, se llama E-UTRAN². La red encargada de distribuir la voz mediante VOIP o conmutación de paquetes desde la E-UTRAN hacia redes privadas, internet o sistemas multimedias, llamados PDNs ³ se llama red troncal EPC ⁴.

Para cada uno de los elementos de la red descritos, existen protocolos de comunicación que para la capa física se encuentran en la interfaz de radio y hacen posible comunicar el UE con la eNB.

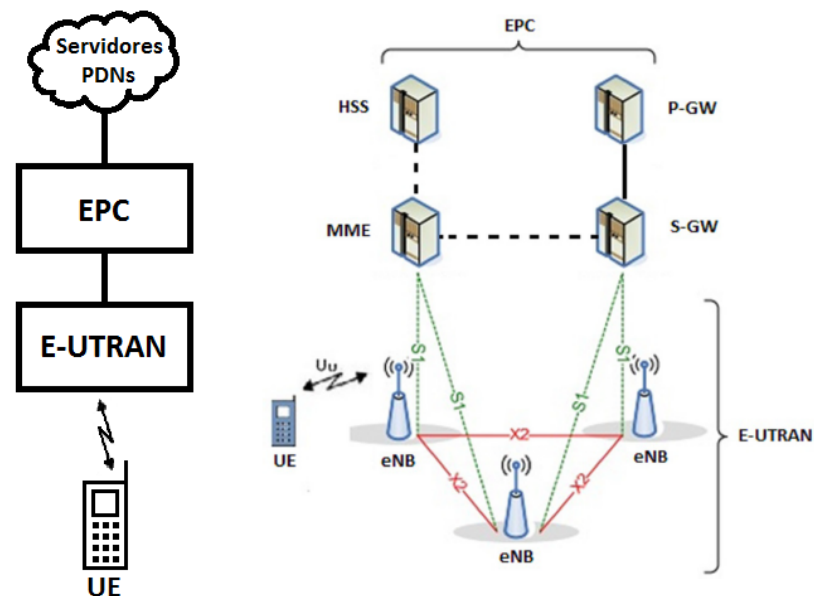


Figura 2.1: Arquitectura de la red 4G LTE. [3]

¹UE: User Equipment, teléfono celular del usuario

²Evolved UTRAN: *Evolved UMTS Terrestrial Radio Access Network*

³Packet Data Network

⁴Evolved Packet Core

2.2.1. Núcleo de Paquete EPC

Dentro de la arquitectura del EPC (*Evolved Packet Core*), en donde está presente la capa física de la red, se encuentran elementos que son los encargados de poseer la información de los abonados de la red HSS (*Home Subscriber Server*), el P-GW (*PND Gateway*) que es el enlace con las redes privadas o PNDs, cada vez que el UE se enciende, se le asigna un P-GW para que posea conectividad siempre. El S-GW (*Serving Gateway*) que envía los datos entre el eNB y el P-GW, actuando de esta manera como un *router*, si aumenta el número de S-GW, aumenta el tráfico en la red. La MME (*Mobile Management Entity*), que envía mensajes de señalización a los UE, si aumenta el número de UE aumentaría el número de MME. De esta manera permite una mayor escalabilidad a la red.

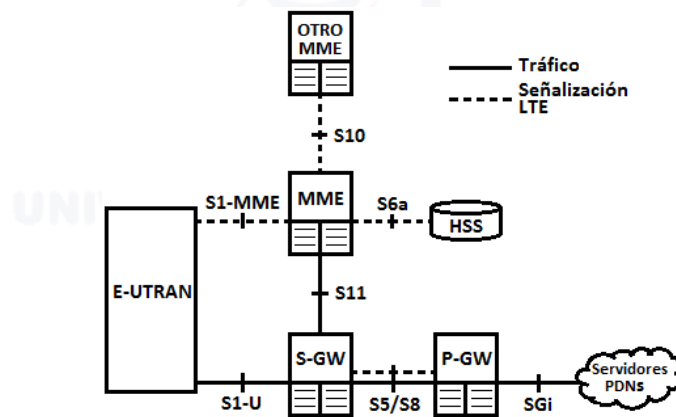


Figura 2.2: Arquitectura de un EPC de la red 4G LTE. [3]

2.2.2. Interfaz Aire

Dentro de la interfaz de aire se encuentra la capa física de la red, en la cual se incorpora la detección y corrección de errores, modulación y codificación, técnicas de transmisión OFDM y SC-FDMA, y las técnicas de múltiple antena (MIMO); para terminar se encuentra el procesador analógico, que cumple la función de convertir la señal digital en señal analógica y transmitir la señal mediante radiofrecuencia.

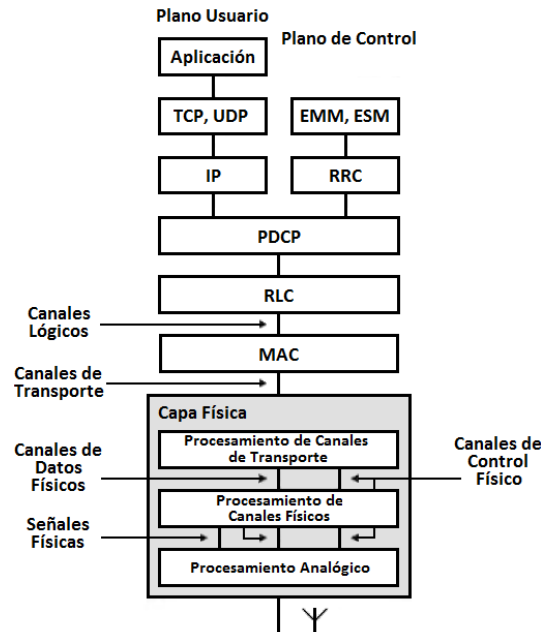


Figura 2.3: Protocolos de la interfaz aire en LTE. [9]

2.3. Capa Física Red 4G LTE

Las principales mejoras que presenta el estándar LTE, se encuentran en la capa física, en donde además de los procesos de detección y corrección de errores, modulación y demodulación, se incluyeron tecnologías como OFDM (*Orthogonal Frequency Division Multiple Access*) para el enlace descendente y SC-FDMA (*Single Carrier Frequency Division Multiple Access*) para el enlace ascendente.

Algunas de las ventajas que presenta el uso de estas tecnologías son, la alta eficiencia espectral y la capacidad de adaptación para la transmisión de datos de banda ancha, la resistencia a la interferencia entre símbolos causada por desvanecimiento por múltiples trayectorias, entre otras.

Estas tecnologías también permiten a LTE proporcionar una variedad de bandas de frecuencia, y un conjunto de anchos de banda escalables. Así como también proporciona un tamaño de trama por debajo de 10 ms lo cual ayuda a minimizar la latencia.

Se incluye técnicas MIMO, es decir se utilizan múltiples antenas tanto en la transmisión como en la recepción. El uso de estas técnicas ayuda a cumplir los requerimientos del estándar sobre todo en lo que refiere a tasas máximas de datos y rendimiento del sistema, obteniendo así fiabilidad en el enlace de comunicación. Los algoritmos MIMO utilizados, se dividen en dos categorías: diversidad de transmisión y multiplexación espacial.

Este tipo de técnicas MIMO ayudan a aumentar la velocidad de datos de un enlace. Las velocidades de datos pueden ser mejoradas linealmente proporcional al número de antenas de transmisión que se tengan.

Los estándares LTE especifican los espectros de radio disponibles en diferentes bandas de frecuencia. Uno de los objetivos de la norma LTE es una integración con los sistemas móviles anteriores.

La capa física es la encargada de procesar toda la información antes de ser enviada a las capas superiores y poder ser transmitida, específicamente, proporciona servicios a la capa MAC en términos de multiplexación y de-multiplexación de los canales de transporte. Envía y recibe bloques de transporte desde y hacia la capa MAC cada 1 *ms* (TTI). Los bloques de transporte generan las palabras de código (*codewords*) durante el proceso de codificación. Estos procesos se ejecutan en los canales físicos de los enlaces ascendente y descendente.

La capa física del enlace descendente (*Downlink*), se representa por el canal DL-SCH (*Downlink Shared Channel*), el mismo que se encarga de la transferencia de datos mediante canales, los datos transferidos pueden ser de información, control, sincronización, entre otros. El principal de estos canales es el PDSCH (*Physical Downlink Shared Channel*), que es el canal físico portador de datos de información, los canales restantes se describirán en los capítulos siguientes, en estos canales se realizan los procesos mencionados anteriormente.

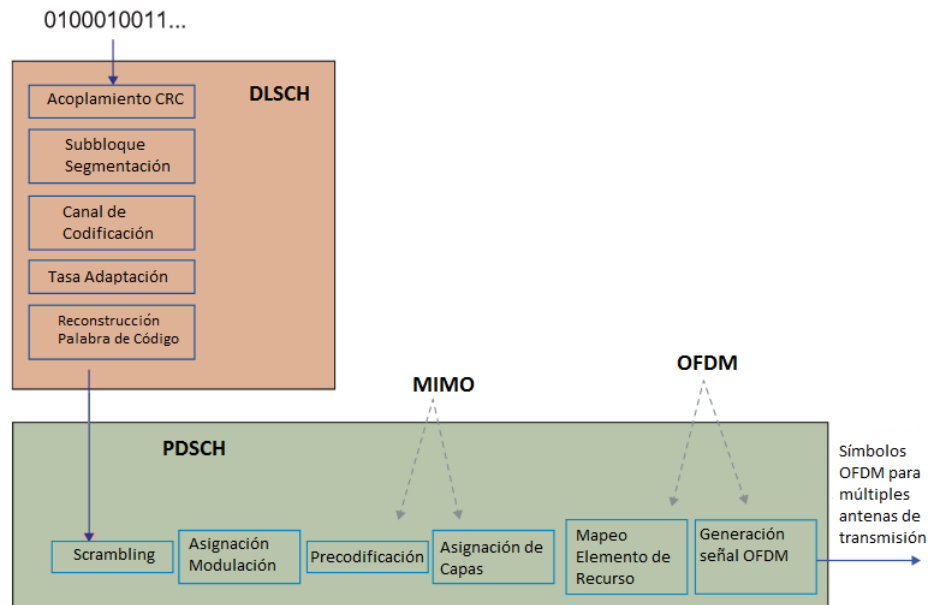


Figura 2.4: Procesos de la capa física LTE para el enlace descendente. [16]

La Figura 2.4 presenta la combinación de procesamiento de la señal aplicada, para el transporte de bloques entregados a la capa física hasta que la señal OFDM se transporta a las antenas para la transmisión, los procesos detallados en la norma 3GPP para enlace descendente se mencionan a continuación:

- Verificación de redundancia cíclica.
- Codificación de Canal.
- Detección y corrección de errores.
- Entrelazamiento de Canales.
- Aleatorización (*Scrambling*).
- Modulación.
- Capa de mapeo y pre-codificación: MIMO.
- Generación de señal OFDM.

La capa física del enlace ascendente (*Uplink*), se representa por el canal UL-SCH (*Uplink Shared Channel*), el mismo que se encarga de la transferencia de datos mediante canales, los datos transferidos pueden ser de información, control, sincronización, entre otros. El principal de estos canales es el PUSCH (*Physical Uplink Shared Channel*), que es el canal físico portador de datos de información, los canales restantes se describirán en los capítulos siguientes, en estos canales se realizan los procesos mencionados anteriormente.

- Verificación de redundancia cíclica.
- Codificación de Canal.
- Detección y corrección de errores (ARQ).
- Aleatorización (*Scrambling*).
- Modulación.
- Transformada inversa de Fourier
- Capa de mapeo y pre-codificación: MIMO.
- Generación de señal SC-FDMA.

En la Figura 2.5 se puede observar la secuencia mencionada, en donde se realiza la combinación del procesamiento de la señal aplicada para el transporte de bloques entregados a la PHY (Capa física de LTE) hasta que la señal SC-FDMA se transfiere a las antenas para la transmisión.

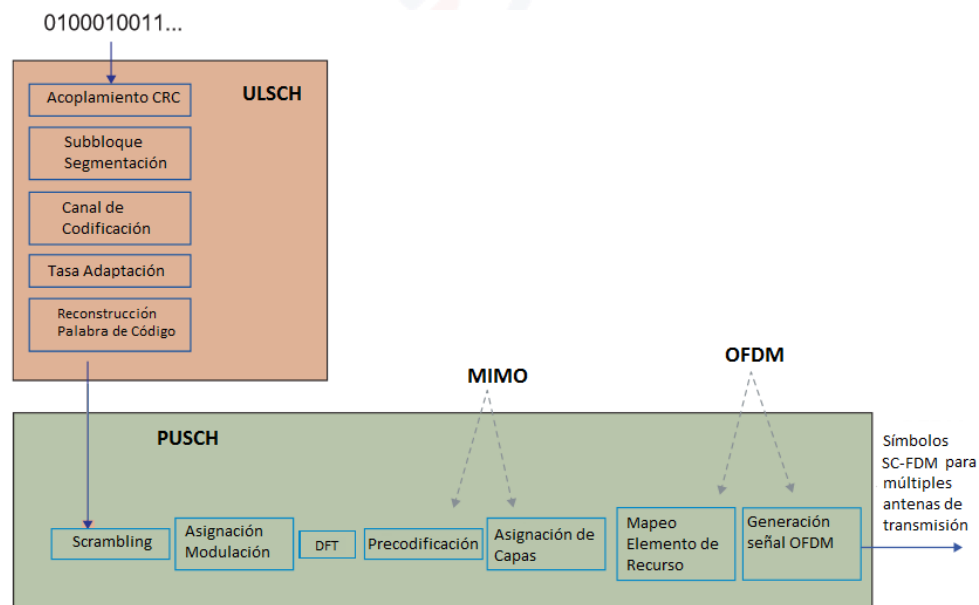


Figura 2.5: Procesos de la Capa física LTE para el enlace ascendente. [16]

2.3.1. Procesamiento del Canal de Enlace Descendente

En la Figura 2.6, se visualizan los bloques de procesamiento para el enlace descendente, los mismos que se aplican a los canales: DL-SCH, PCH, MCH y BCH; en donde se puede enviar uno o dos bloques de transporte (TB) en cada TTI, y la codificación de la palabra de código a los canales PDCCH, PCFICH y PHICH.

La función del identificador temporal de la red de radio (RNTI), es identificar al destinatario en la red, mientras que el identificador de celda, identifica la celda transmisora.

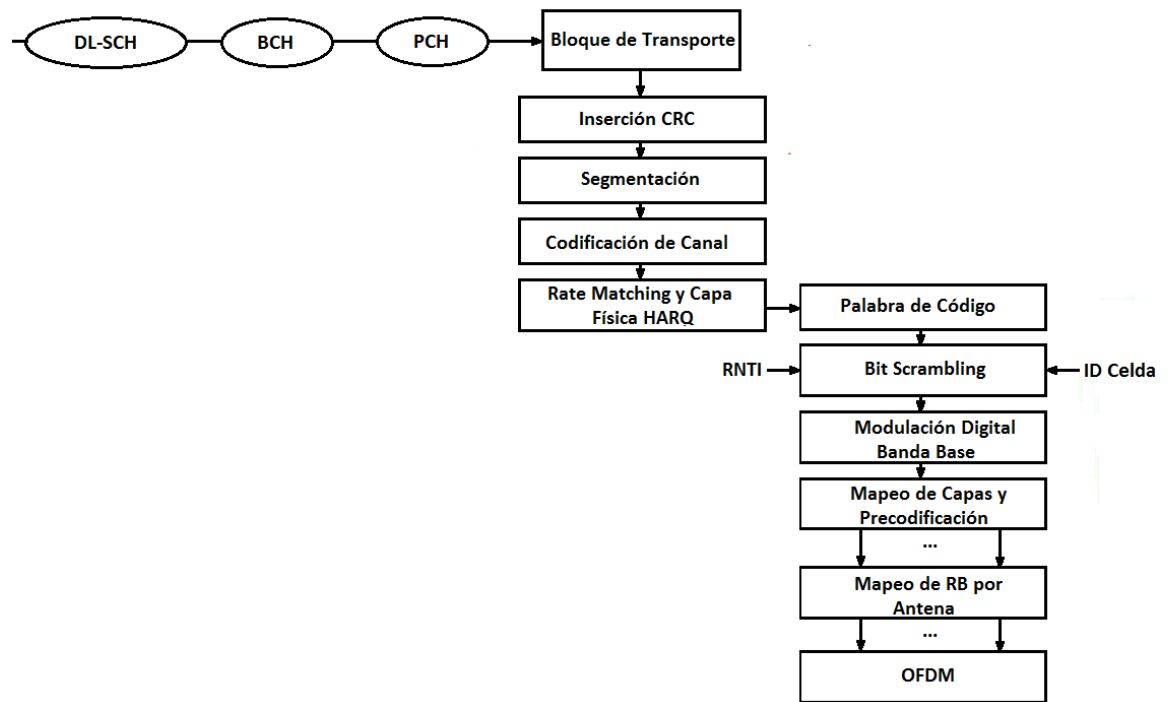


Figura 2.6: Procesos de los canales físicos enlace descendente. [2]

Una breve descripción de los canales físicos se presenta a continuación:

- **PBCH:** Canal físico de difusión, transmite información específica de la celda.
- **PDSCH:** Canal físico compartido de enlace descendente, transmite los servicios del plano de control, así como los servicios de datos del usuario.
- **PDCCH:** Canal físico de control del enlace descendente, transmite información de control.
- **PHICH:** Canal físico indicador de ARQ Híbrida, lleva la información ACK o NACK recibidas por el eNB.
- **PCFICH:** Canal físico indicador de formato de control, lleva la información del formato de control de la subtrama actual hacia el UE.
- **PMCH:** Canal físico de multidifusión, transmite los servicios de multidifusión del plano de usuario.

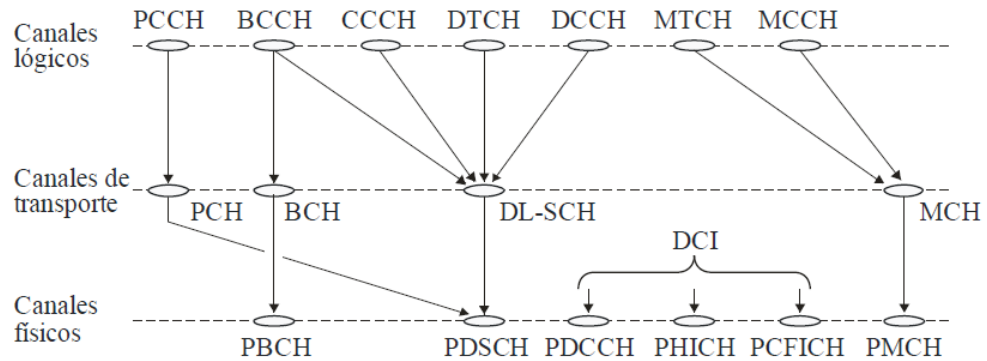


Figura 2.7: Canales del enlace descendente.[9]

La capa MAC entrega uno o dos bloques de transporte cada 1 *ms*. El proceso de CRC adjunto, se realiza dependiendo del canal. Para los canales PDCCH y PBCH, se realiza un CRC de 16 bits, mientras que para el canal DL-SCH se realiza un CRC de 24 bits.

La segmentación del código de bloque, se realiza en caso de que el tamaño del bloque de transporte sobrepase los 6144 bits. Este proceso consiste en dividir o segmentar los bloques de transporte en bloques de código, Figura 2.8, en donde los bloques de código son de longitud menor o igual a 6144 bits.

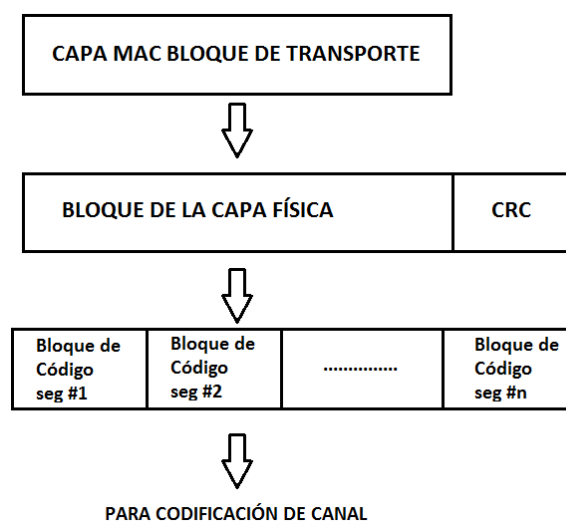


Figura 2.8: Procesamiento hacia la codificación del canal. [2]

Una vez obtenidos los códigos de bloque, se ingresan uno a uno al proceso de codificación de canal, Figura 2.9, para el enlace descendente se utilizan dos esquemas de codificación para los canales de transporte, la codificación turbo (DL-SCH, PCH y MCH) y los códigos convolucionales *Tail Biting* (PBCH), los dos esquemas con una tasa $R = 1/3$. Para cada bloque de código se agrega 24 bits de CRC, antes de la codificación.

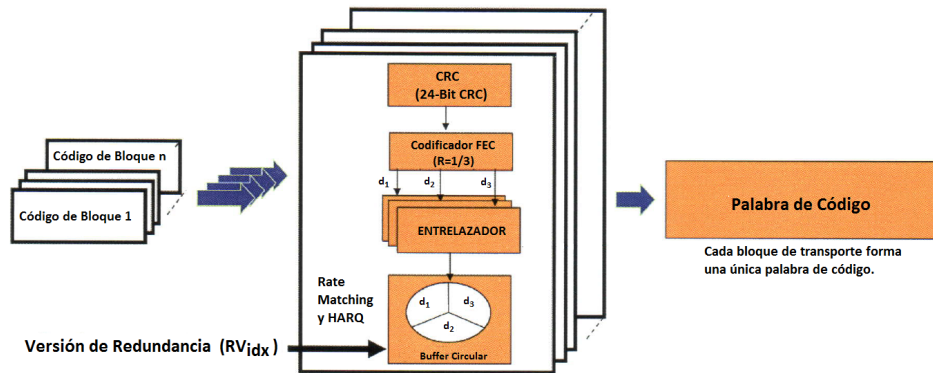


Figura 2.9: Codificación del canal de enlace descendente. [2]

El codificador turbo, es un código convolucional concatenado paralelo (PCCC), constituido por dos codificadores convolucionales de 8 estados con una tasa de codificación de $1/2$. El codificador turbo posee un entrelazador interno, Figura 2.32, con una tasa de codificación $1/3$. Están definidos 188 tamaños de bloques, Tabla 3.3, según la normativa 3GPP.

Del proceso de codificación se obtienen tres flujos de datos d_0 , d_1 y d_2 que se introducen en el entrelazador del proceso de *rate matching* explicado mas adelante. El entrelazador es un intercalador de fila a columna. La salida se recoge en un *buffer* circular utilizado para la adaptación de velocidad o *rate matching*, la misma que se realiza basándose en la versión de redundancia seleccionada indicada por RV_{idx} , que puede ser 0,1,2 o 3.

Después de realizar el proceso de *rate matching*, la salida de cada bloque de código se multiplexa en serie conjuntamente, dando como resultado una palabra de código. Puede haber hasta dos palabras de código si dos bloques de transporte se transmiten simultáneamente a través de múltiples antenas.

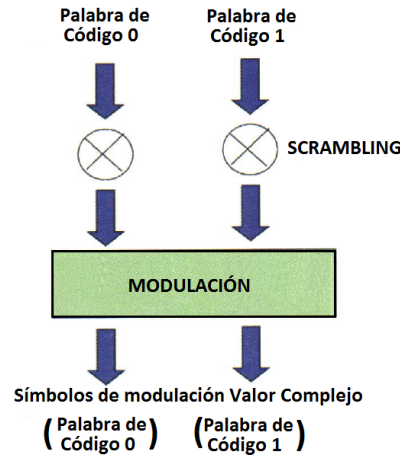


Figura 2.10: Aleatorización (*Scrambling*) y modulación. [2]

Los bits codificados se envían como uno o dos flujos (palabra de código 0 y palabra de código 1) correspondientes a uno o dos bloques de transporte, Figura 2.10. Las palabras de código son bits codificados con secuencia binaria pseudo-aleatoria. La secuencia binaria se inicializa con la identidad de la celda para hacerla específica para cada celda. El valor de inicialización de PDSCH se basa en la identidad de celda y el RNTI asignado al UE, para crear una secuencia específica para cada UE.

Los bits codificados se asignan a símbolos de modulación usando uno de los siguientes esquemas de modulación: BPSK, QPSK, 16-QAM o 64-QAM. En el canal PDSCH se utilizan las modulaciones QPSK, 16-QAM y 64-QAM.

Una vez realizada la modulación, se procede a realizar el mapeo de capas, los símbolos de modulación se asignan a un número dado de capas: Para la transmisión de un solo puerto de antena, sólo se utiliza una capa. Para la multiplexación espacial, 1 a 4 capas representan el número de flujos de datos paralelos.

Tabla 2.1: Mapeo de capas multiplexación espacial. [2]

Número de Capas	Número de Palabras de Código	Palabra de Código para el mapeo de capas
1	1	Palabra de código cero, capa cero
2	1	Palabra de código cero, capas cero y uno
	2	Palabra de Código cero, capa cero. Palabra de código uno, capa uno
3	2	Palabra de Código cero, capa cero. Palabra de código uno, capas uno y dos
4	2	Palabra de Código cero, capas cero y uno. Palabra de código uno, capas dos y tres.

Para la multiplexación espacial, las capas representan el número de flujos de datos paralelos transmitidos. El número de capas es igual o menor que el número de puertos de antena. Para realizar la asignación de capas, las palabras de código se dividen en capas. La asignación de palabras de código a las capas se muestra en la Tabla 2.1.

Tabla 2.2: Mapeo de capas diversidad de transmisión. [2]

Número de Capas	Número de Palabras de Código	Palabra de Código para el mapeo de capas
2	1	Palabra de código cero, capas cero y uno
4	1	Palabra de código cero, capas cero, uno, dos y tres

Para la diversidad de transmisión, Tabla 2.2, la asignación de capas es igual al número de puertos de antena. Sólo hay una palabra de código, en donde para una configuración de puerto de dos antenas, la palabra de código se asigna a dos capas, en el caso de 4 antenas la palabra de código se asigna a cuatro capas. El orden de diversidad es siempre 2, ya que hay dos flujos redundantes para cada símbolo.

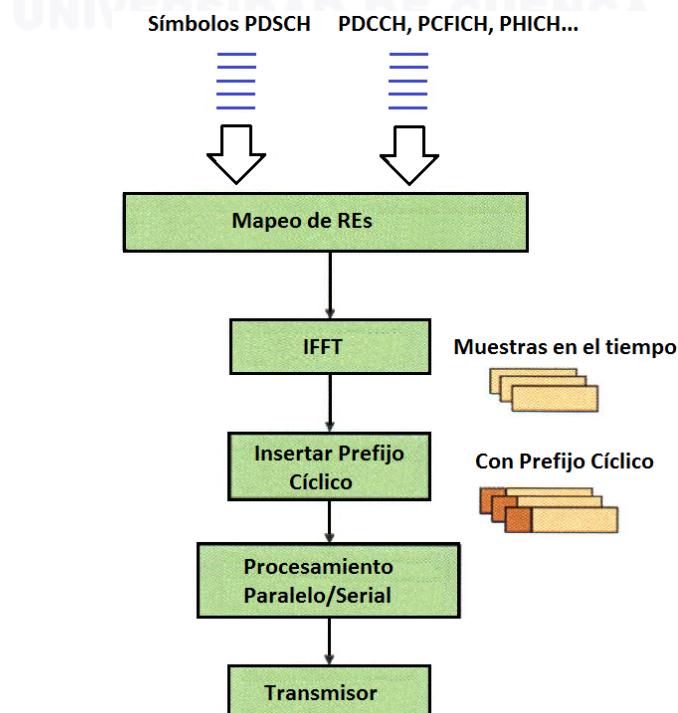


Figura 2.11: Modulación OFDM. [2]

Los símbolos precodificados de PDSCH se acumulan y se asignan a los elementos de recursos en puertos de antena seleccionados. Todos los canales, incluidos los canales de control, que se transmiten en una subtrama, también son asignados a los elementos de recursos. La asignación de elementos de recurso convierte símbolos de flujos en serie a flujos paralelos y los asigna a diferentes subportadoras para cada símbolo OFDM.

Para cada tiempo de símbolo OFDM, la operación IFFT se realiza para crear muestras de dominio de tiempo OFDM para todas las subportadoras. El prefijo cíclico se inserta antes y después de cada *slot*, en donde para el primer *slot* es un poco más grande, lo que permite que el prefijo cíclico sea igual en el resto de los *slots*, lo cual permite disminuir la interferencia intersímbolo (ISI). Después de aplicar el prefijo cíclico, los *slots* de tiempo se convierten de paralelo a serie y se transmiten a través del ancho de banda del canal.

A continuación se describe como se generan los canales, (PDSCH, PBCH, PCFICH, PHICH, PDCCH) así como las señales que abarcan el enlace descendente (RS, PSS, SSS).

Canal PDSCH

En el canal físico compartido de enlace descendente (PDSCH) se pueden transmitir uno o dos bloques de transporte durante cada TTI (1 *ms*), en donde se puede enviar de una a dos palabras de código. En la Figura 2.12, se observa que para cada bloque de transporte existe un CRC de 24 bits. El bloque de transporte se divide en bloques de código de 6144 bits. Se realiza una codificación turbo con una tasa de 1/3. Los flujos de datos, d_0 , d_1 , y d_2 se intercalan usando un entrelazador de fila a columna, y esta salida ingresa en un *buffer* circular para cada proceso HARQ en el UE. El algoritmo de ajuste de velocidad selecciona bits del *buffer* circular según el parámetro de la versión de redundancia. Existen cuatro versiones de redundancia diferentes ($rv = 0, 1, 2, 3$). Una vez que se adapta la tasa de velocidad los datos se concatenan en serie.

La palabra de código se codifica mediante una secuencia *scrambling*, la misma que se basa en el ID de la celda y RNTI del usuario. Los bits codificados se modulan mediante los esquemas: QPSK, 16-QAM o 64-QAM. La selección se basa en la retroalimentación CQI del equipo de usuario.

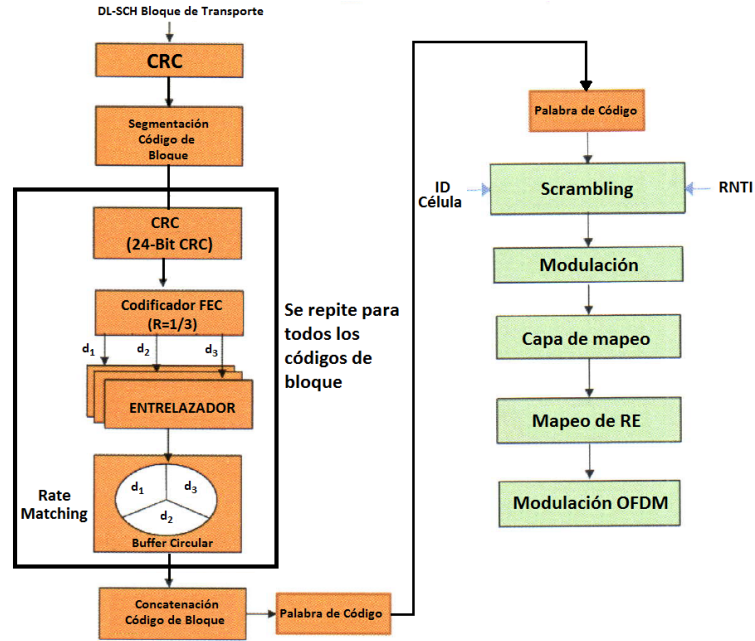


Figura 2.12: Generación del canal PDSCH. [2]

Las palabras de código moduladas se asignan a capas; en diversidad de transmisión el número de capas es igual al número de puertos de antena, mientras que en multiplexación espacial se utiliza de 1 a 4 capas, igual o menor que el número de puertos de antena.

Los símbolos modulados se asignan a los elementos de recurso (k, l) en los bloques de recurso asignados. El mapeo se realiza primero a lo largo de la frecuencia, y segundo a lo largo de los símbolos. Los símbolos de modulación se mapean incrementando el índice de subportadora (k) primero a lo largo de los RB asignados, hasta llegar al final del bloque de recurso, y se incrementa el índice de símbolos (l) .

Para calcular la capacidad bruta del canal PDSCH, es necesario conocer el número de RE, y el número de niveles de modulación M , entonces:

$$CB = \frac{\left(\frac{RE \times M}{10}\right)}{1000}$$

Se divide para 1000, para obtener en Mbits/s.

Canal PBCH

El canal físico de *broadcast* PBCH lleva el bloque de información maestro (MIB), que contiene los parámetros más importantes del sistema, como el ancho de banda del canal, en este canal se realiza un CRC de 16 bits el mismo que se codifica de acuerdo a la configuración del eNodeB.

PBCH utiliza la codificación *Tail Biting*, en donde los bits codificados se encuentran entrelazados y con una tasa de velocidad similar. Para un prefijo cíclico normal, el número de bits es igual a 1920, y para un prefijo cíclico extendido es 7728. En este canal se modula mediante el esquema QPSK, en donde los símbolos son asignados mediante dos configuraciones, un solo puerto de antena, o mediante diversidad de transmisión para 2 o 4 puertos de antena. Estos símbolos son mapeados a cada elemento de recurso asignado para el canal PBCH.

PBCH se encuentra en el segundo *slot* (*slot* uno) de la trama. Ocupa los primeros cuatro símbolos de este *slot* en el dominio del tiempo, mientras que en el dominio de la frecuencia, ocupa 6 RBs o 72 subportadoras, Figura 2.13. El equipo de usuario puede decodificar rápidamente el PBCH sin tener que conocer el ancho de banda del canal. La información de ancho de banda de canal se lleva en términos de número de RB.

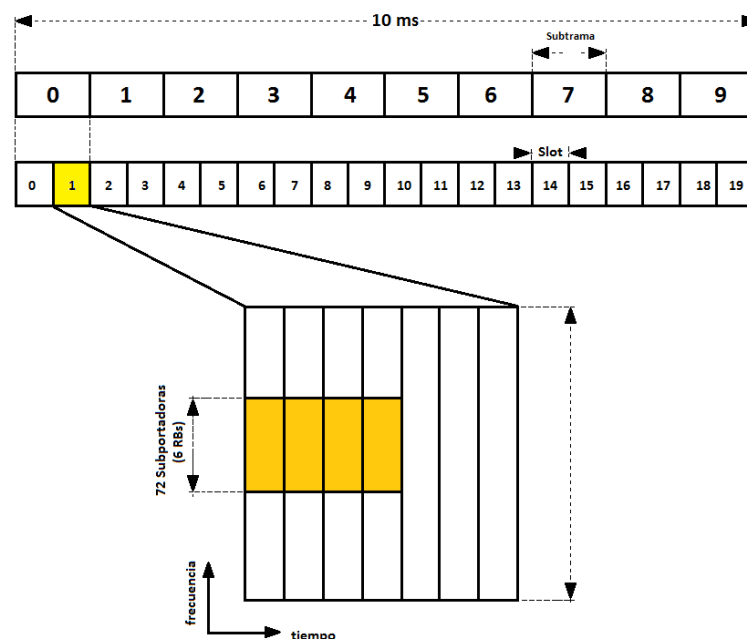


Figura 2.13: Mapeo del canal PBCH. [2]

Canal PCFICH

El canal físico indicador de formato PCFICH, indica el número de símbolos OFDM en una subtrama del canal de control PDCCH, los mismos que pueden ser entre 1 y 4, dependiendo del ancho de banda del canal. Si el número máximo de bloques de recurso es superior a 10, el número de símbolos está entre 1 y 3; y por el contrario si es menor o igual que 10, el número de símbolos oscila entre 2 y 4.

El indicador de formato de control (CFI) son dos bits, que indican 1, 2, 3 símbolos del canal de control, esto para el caso que el número máximo de bloques de recurso sea superior a 10. Si el número de bloques de recurso es menor o igual a 10, el número de símbolos de canal de control es $CFI + 1$, esto se resume en la Tabla 2.3.

Tabla 2.3: Número de símbolos de control. [2]

CFI	Número de Símbolos (RBs > 10)	Número de Símbolos (RBs ≤ 10)
1	1	2
2	2	3
3	3	4
4	reservado	reservado

El CFI se codifica con una tasa de 1/16, es decir con un código de bloque (32, 2). Los 32 símbolos de código se codifican mediante una secuencia aleatoria que se inicia con el ID de la celda, y el código resultante se aplica modulación QPSK, quedando 16 símbolos de valor complejo. El mapeo de capas se realiza de la misma manera que el canal PBCH, y los 16 símbolos de modulación se asignan a cuatro grupos de elementos de recursos, en el primer símbolo del canal de control. En la Figura 2.14 se observa el proceso antes descrito.

Canal PHICH

En el canal físico indicador de HARQ; los indicadores PHICHs, se mapean en grupos, los mismos pueden ser cuatro u ocho indicadores, dependiendo de la configuración del prefijo cíclico, 4 para el prefijo cíclico normal.

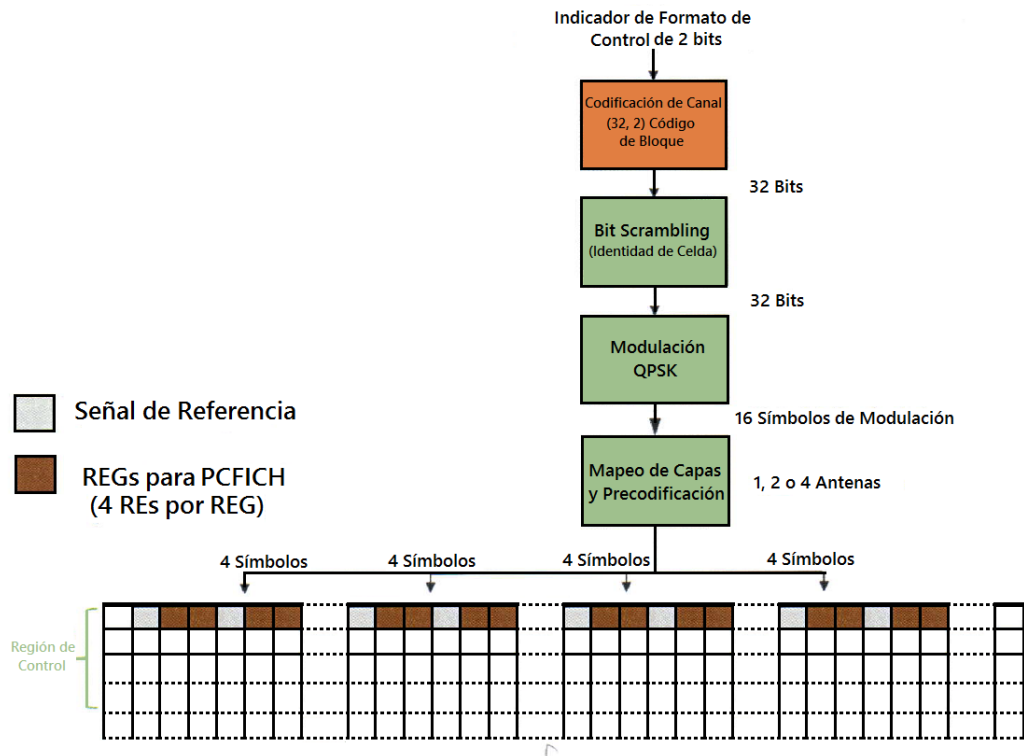


Figura 2.14: Generación del canal PCFICH. [2]

El indicador HARQ (HI), es un bit; indica 1 para ACK y 0 para NAK. El HI está configurado con codificación de 1/3; este resultado se codifica con una secuencia aleatoria que se inicia con el ID de celda, y estos se propagan ortogonalmente. El número de bits después de la propagación es de 12 para un código de cuatro bits.

Los bits de propagación se modulan con el esquema BPSK, y los símbolos modulados se asignan a capas, dependiendo de la longitud del prefijo cíclico y del número de puertos de antena. El indicador PHICH se transmite en el mismo conjunto de puertos de antena que el PBCH, utilizando la transmisión diversificada.

Los símbolos de modulación precodificados en cada puerto de antena se asignan a los grupos de elementos de recurso. Los 12 símbolos se asignan a tres REG asociados con el grupo PHICH. La ubicación inicial del primer REG del grupo PHICH se deriva de la identidad de la celda. Los otros dos grupos están equidistantes del primer grupo.

Solicitud de Repetición Automática Híbrida HARQ

Dentro de los sistemas inalámbricos, se realiza la verificación de que los datos lleguen correctamente en el receptor, caso contrario, se pide un reenvío de datos. Si los datos llegaron correctamente, el receptor envía una señal ACK, en caso de que los datos lleguen con error el receptor envía una señal NACK, solicitando la retransmisión de los datos.

El método empleado en la red 4G LTE, es la solicitud de repetición automática Híbrida HARQ, que consiste en pedir un reenvío de datos, mientras almacena temporalmente los datos recibidos con error en un buffer, para realizar una suma con los nuevo datos recibidos, para poder reconstruir el mensaje, se permite hasta 4 solicitudes de reenvío de datos.

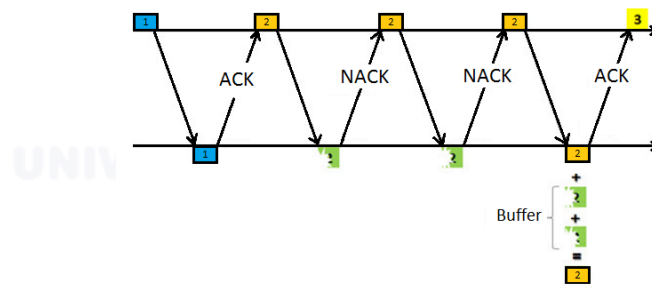


Figura 2.15: Solicitud de repetición automática híbrida. [13]

Canal PDCCH

En el canal de control de enlace físico descendente (PDCCH), se transmite la información de acuse de recibo HARQ, para la transmisión de enlace descendente, y su vez transmite los indicadores de aviso. El PDCCH, utiliza los 2, 3, o 4 primeros símbolos de cada subtrama, ocupando un ancho de banda de 1.4 MHz, y los 1, 2 o 3 primeros símbolos de subtrama para una anchura mayor de 1.4 MHz.

Esta información se consolida en el mensaje de control de información descendente DCI, el mismo que se coloca en uno o varios elementos de control CCE. Los elementos de control constan de 9 grupos de elementos de recurso (REG), en donde cada REG posee 4 elementos de recursos, los mismos son utilizados para asignar los RB, ajustar la codificación, control de potencia, y revisar la redundancia HARQ.

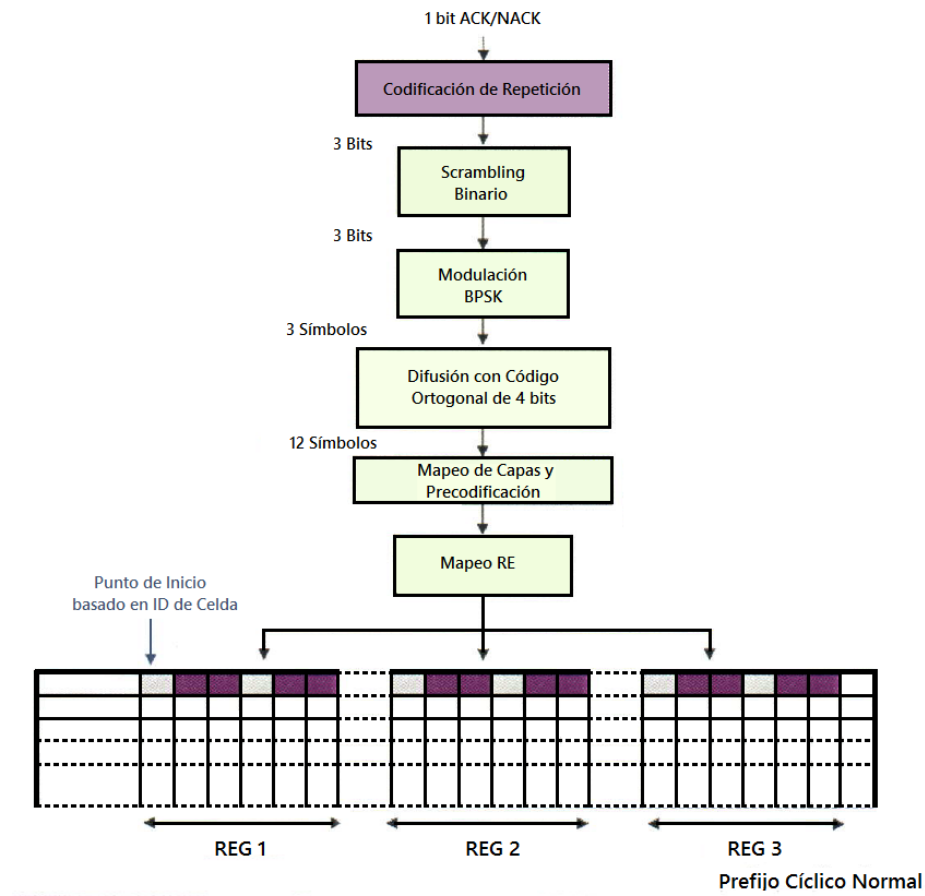


Figura 2.16: Generación del canal PHICH. [2]

Según el formato que utilice PDCCH, existen 1, 2, 4, 8, elementos de control, los mismos se pueden visualizar en la Tabla, 2.4.

Tabla 2.4: Formato PDCCH. [2]

Formato	CCE	REG	Símbolos	Bits
0	1	9	36	72
1	2	18	72	144
2	4	36	144	288
4	8	72	288	576

Las Figuras 2.17 y 2.18 muestran como están posicionados los distintos canales en el enlace descendente.

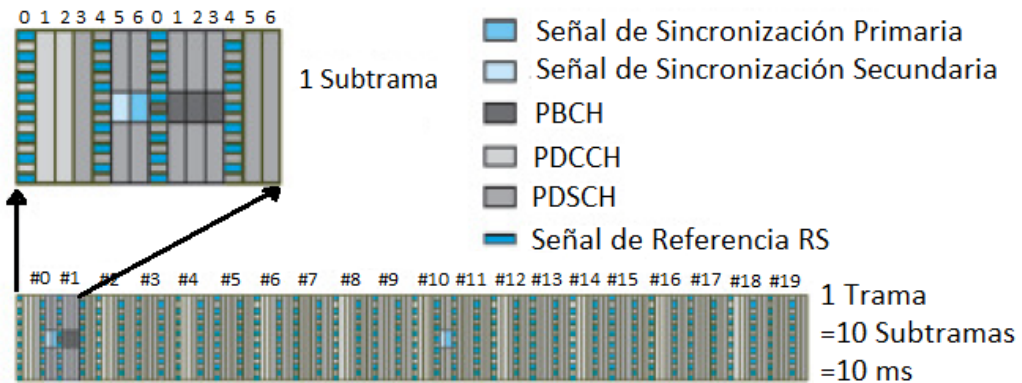


Figura 2.17: Posición de los canales físicos del enlace descendente. [6]

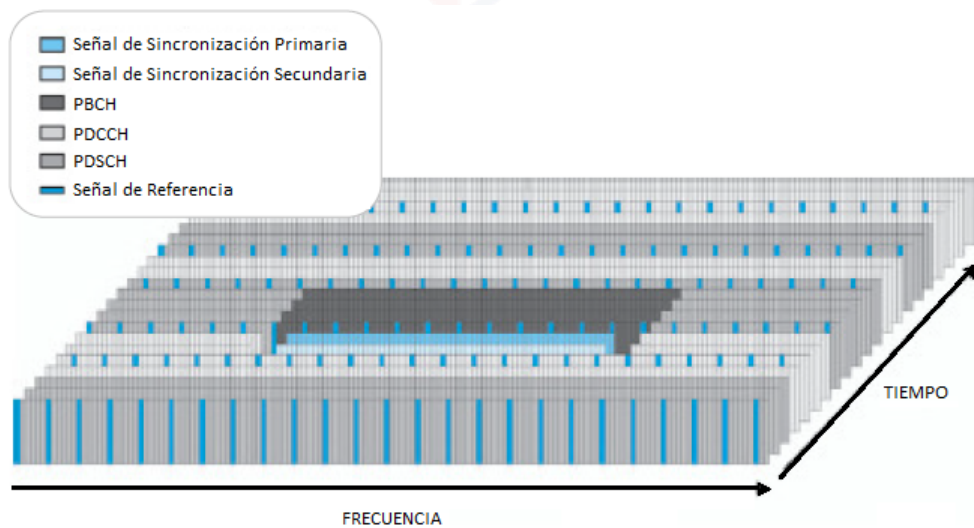


Figura 2.18: Tiempo y Frecuencia de los canales físicos enlace descendente. [6]

Señales Físicas del Enlace Descendente

Las señales físicas están presentes en los dos enlaces, ascendente y descendente, las mismas que llevan información para identificar la celda, sincronización temporal y estimación del canal utilizado.

Para el enlace descendente existen 3 señales físicas, en donde la señal PSS (*Primary Synchronization Signal*), señal de sincronización primaria, es la que se encarga de la temporización del intervalo y obtiene una parte de la identidad

de la celda. La señal SSS (*Secondary Synchronization Signal*) señal secundaria de sincronización, es la encargada de obtener la identidad de la trama así como la temporización de la misma. Y finalmente la señal RS (*Reference Signal*), señal de referencia, estima la función de transferencia del canal descendente.

A continuación se describe mas a detalle cada una de las señales físicas mencionadas anteriormente.

- **Señal física PSS**

La señal PSS es transmitida en el penúltimo símbolo OFDM de los *slots* 0 y 10 y se transmite dos veces por trama sobre las mismas subportadoras. Las subportadoras reservadas son 72, es decir 6 RB, pero estas dos señales ocupan siempre 62 subportadoras centrales del canal utilizado por LTE.

Se aplica una modulación secuencial en el dominio de la frecuencia a las subportadoras, es decir poseen una diferente fase, pero la misma potencia.

La señal PSS se utiliza para identificar la temporización, centro de banda, y la identidad de la célula (3 posibles).

- **Señal física SSS**

La señal SSS es transmitida en el último símbolo OFDM de los *slots* 0 y 10, y se transmite dos veces por trama sobre la misma subportadora. Las subportadoras reservadas son 72, es decir 6 RB, pero estas dos señales ocupan siempre 62 subportadoras centrales del canal utilizado por LTE. Se utiliza para identificar el grupo de identidades de celda, dentro de los 168 grupos posibles, pues existen 504. Estas señales son para sincronizar el UE, el mismo que una vez realizada la sincronización analiza la respuesta del canal, amplitud y fase mediante la RS.

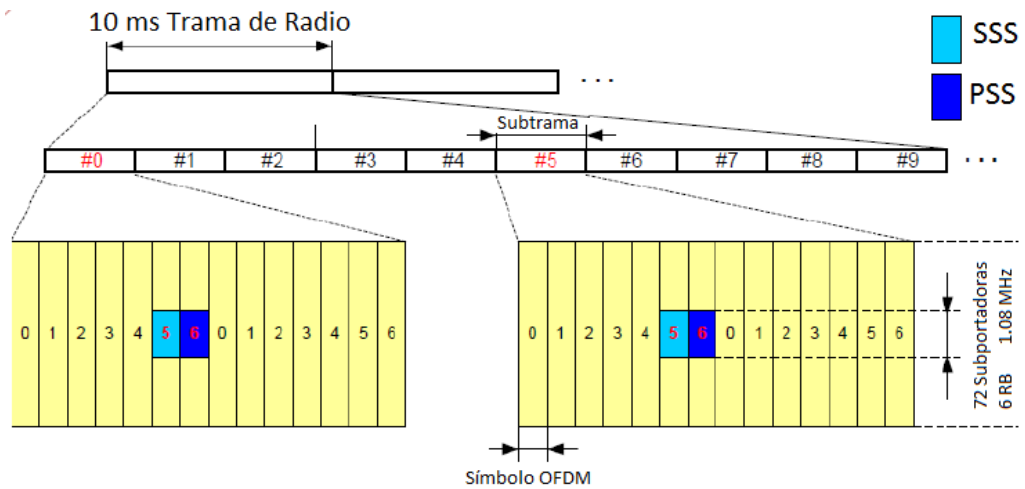


Figura 2.19: Posición señales físicas PSS y SSS. [2]

- Señal física RS

Las señales de referencia que se transmiten identifican los puertos de antena, los mismos que para una configuración básica, es decir un solo puerto por cada RB, se emplean dos subportadoras, por cada 7 símbolos, con una duración de $5ms$; es decir, 4 elementos de recursos. Para el caso de dos puertos, se emplean 4 subportadoras, y se utilizan 8 elementos de recursos y para 4 puertos se requieren 12 elementos de recursos. La separación entre subportadoras en frecuencia es de 75 KHz (5×15). La ubicación de estas señales se puede visualizar en la Figura 2.20. En la cual se observa que las RS se colocan o se transmiten en los símbolos, primero y quinto de cada *slot*, cada 6 subportadoras.

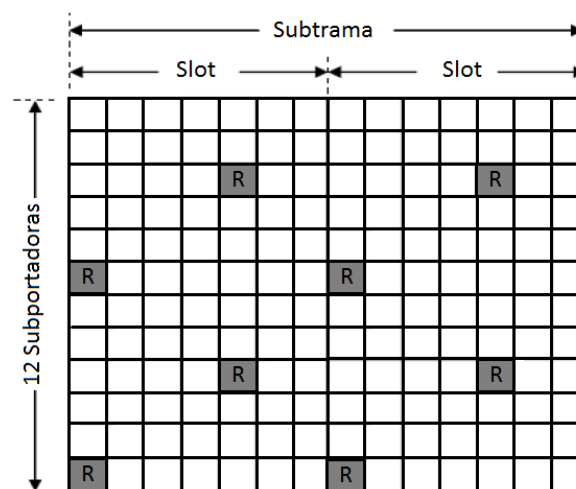


Figura 2.20: Posición señales físicas configuración básica. [6]

2.3.2. Procesamiento del Canal de Enlace Ascendente

Como se observan en la Figura 2.21 para el enlace ascendente los canales físicos son tres:

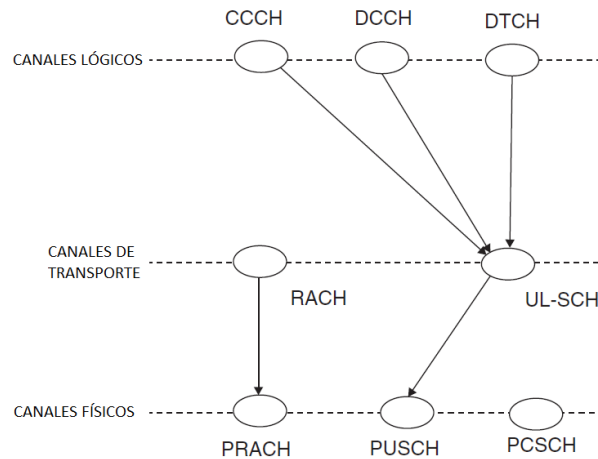


Figura 2.21: Canales del enlace ascendente. [9]

- PRACH: Canal físico de acceso aleatorio, transporta un argumento que se utiliza para activar el procedimiento de acceso aleatorio en el eNB.
- PUSCH: Canal físico compartido del enlace ascendente, en este canal se transporta la información de control para la capa superior, así como los datos del usuario.
- PUCCH: Canal físico de control de enlace ascendente, transporta información de control.

A continuación se describen más a detalle los canales físicos mencionados.

- **PUSCH:** El canal compartido físico de enlace ascendente (PUSCH), transporta información de control del UE y la DMRS, así como los datos de usuario. Utiliza modulaciones QPSK, 16QAM y 64QAM. Los recursos asignados para el enlace ascendente vienen hacia el UE por el DCI.

En la Figura 2.22, se representa como está distribuido el canal PUSCH dentro de una subtrama.

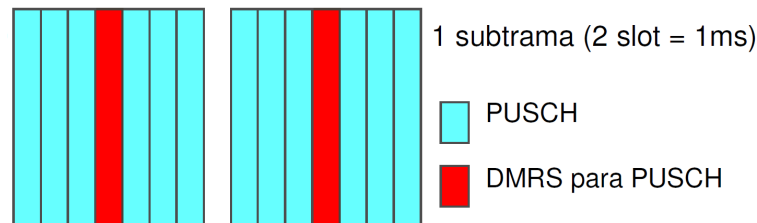


Figura 2.22: Canal físico PUSCH. [6]

- **PUCCH:** El canal de control físico (PUCCH), posee la información de control del equipo de usuario (información ACK/NACK, CQI, indicador de calidad del canal). En la Figura 2.23, se muestra como se distribuye el canal PUCCH dentro de una subtrama.

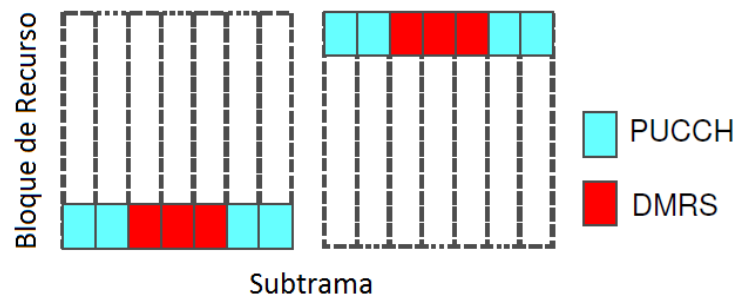


Figura 2.23: Canal físico PUCCH. [6]

- **PRACH:** En el canal físico de acceso, es empleado para dar acceso al UE a la red, así como para realizar una reconexión del UE, para realizar la sincronización por el UE. En el canal PRACH, el UE envía una señal, tras una respuesta de la red positiva, el equipo de usuario, envía su identidad y el motivo por el cual quiere acceder a la red mediante el canal PUSCH. El canal PRACH, tiene un ancho de banda de $1,08\text{MHz}$, que es igual a 6 RB.

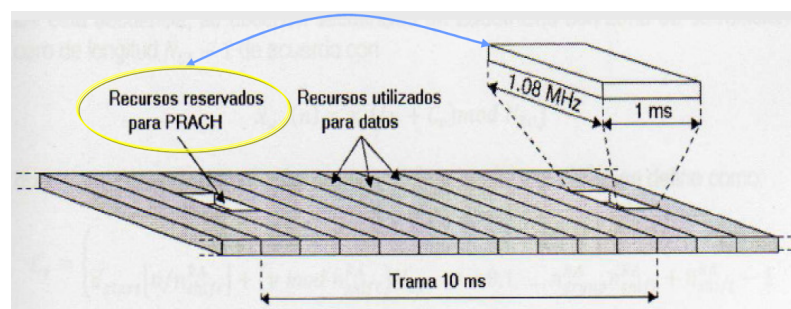


Figura 2.24: Canal físico PRACH. [6]

Indicador de Calidad de Canal CQI.

Un indicador de calidad de canal se encarga de definir la eficiencia que tiene el canal. Para LTE, existen CQI definidos, los mismos se visualizan en la Tabla 2.5.

Señales Físicas de enlace Ascendente.

Para el enlace ascendente existen solamente dos señales; DMRS (*Demodulation Reference Signal*), señal de referencia de demodulación, es la que se encarga de la sincronización y la estimación del canal ascendente. Y la señal SRS (*Sounding Reference Signal*) señal de referencia de sondeo, estima el canal para planificar las frecuencias en el enlace ascendente.

- **DMRS:** La señal de referencia de demodulación (DMRS), es empleada para sincronizar el canal ascendente. Existen dos tipos de señales DMRS, las cuales se encuentran presentes para los canales PUSCH y PUCCH. La señal DMRS/PUSCH es enviada en el cuarto símbolo SC-FDM, en caso del prefijo cíclico normal, para el caso del prefijo cíclico extendido, se envía en el tercer símbolo, de cada RB.

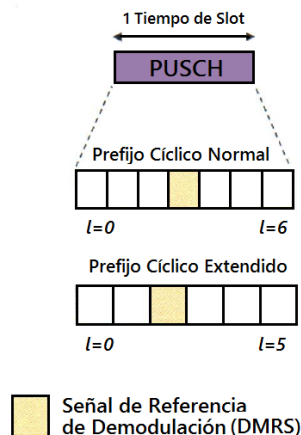


Figura 2.25: Señal de referencia DMRS. [2]

- **SRS:** La señal de referencia de sondeo (SRS) se emplea para estimar el canal ascendente, facilitando la planificación y adaptación del enlace, esta señal es enviada en el último símbolo SC-FDM de la subtrama, sobre subportadoras no utilizadas por el canal PUCCH. Para determinar el número de RE que se utilizan, se debe conocer el ancho de banda del UE (0.,3), y la configuración del ancho de banda en la celda (0.,7).

Procesamiento UL-SCH

El canal de transporte UL-SCH, se asigna al canal físico PUSCH, en donde el procesamiento es parecido al realizado en DL-SCH. La Figura 2.26 muestra el procesamiento del canal UL-SCH, codificación CRC unida al bloque de transporte, se realiza una segmentación en bloques de código de 6144 bits, codificación turbo con una tasa de velocidad igual a $1/3$, y la concatenación del código de bloques para formar la palabra de código.

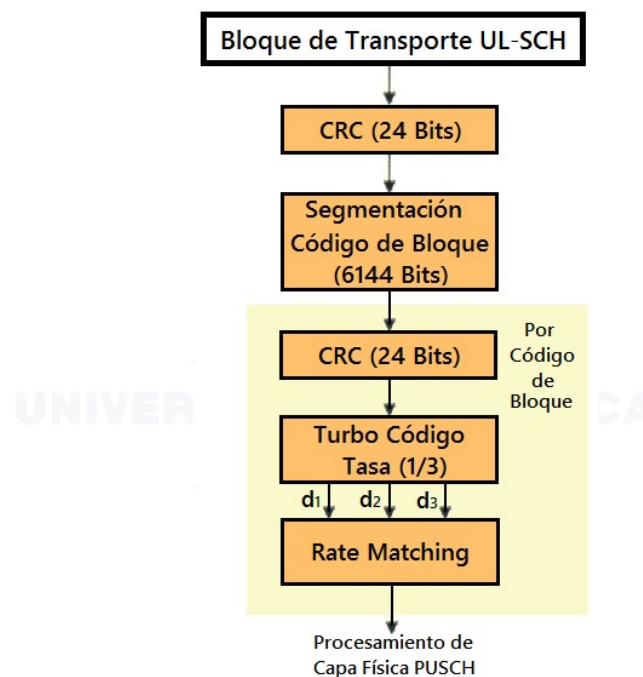


Figura 2.26: Procesamiento UL-SCH. [2]

Procesamiento de datos y control de información PUSCH

El canal PUSCH transporta solamente UL-SCH con información de control adicional, Figura 2.27, esta información incluye el indicador de calidad de canal (CQI), indicador de matriz de precodificación (PMI), indicador de rango (RI) y la señal ACK/NACK.

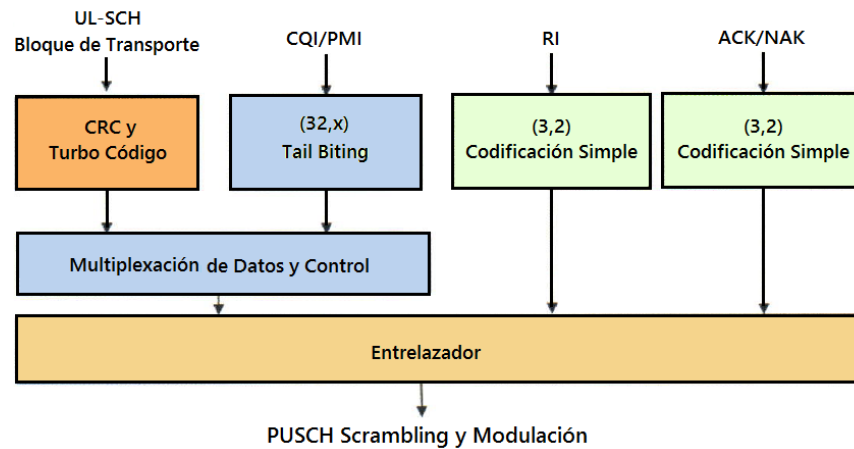


Figura 2.27: Procesamiento PUSCH. [2]

En la Figura, 2.28, se observa como se distribuyen la información adicional, los símbolos CQI se asignan primero, seguidos por los símbolos de datos. Los bits RI ocupan cuatro símbolos alrededor de DMRS. Los símbolos RI se multiplexan con símbolos de datos. Los símbolos del indicador de ARQ híbrido (HI) ACK / NAK se colocan en símbolos de datos alrededor de DMRS.

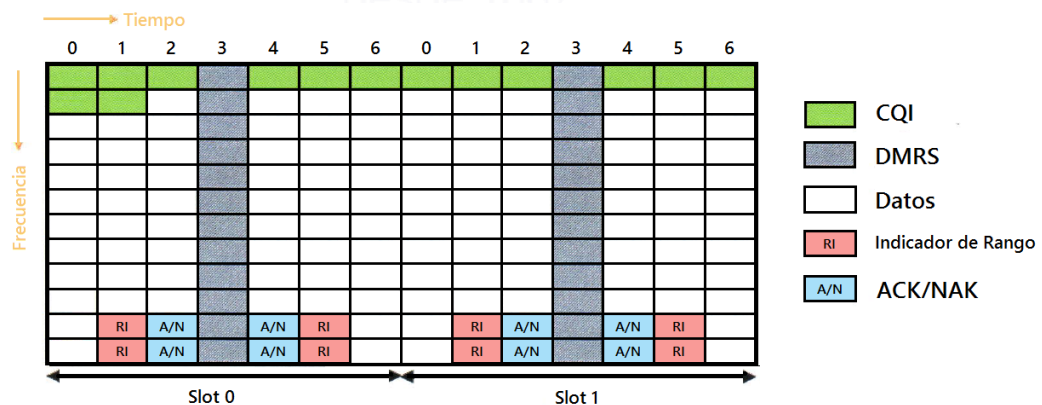


Figura 2.28: Multiplexación de datos y control. [2]

El flujo que siguen los símbolos dentro del canal PUSCH se presenta en la Figura 2.29, en donde la modulación puede ser para datos y control, QPSK, 16-QAM O 64-QAM, mientras que para ACK/NACK se emplea solamente la modulación QPSK. Una vez modulados los símbolos, se procesan con la transformada DFT, generando los símbolos SC-FDMA, y estos símbolos entran en proceso de modulación según la IFFT en donde se agrega el prefijo cíclico configurado.

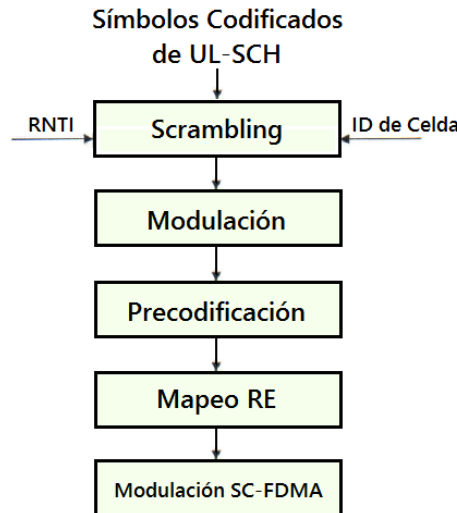


Figura 2.29: Diagrama de procesamiento PUSCH. [2]

2.3.3. Planificación de usuarios

La planificación de usuarios, es la característica más importante de la red LTE, pues permite seleccionar en cada momento los usuarios que se van a conectar, así como la selección de los parámetros de transmisión: la modulación y el esquema de codificación MCS ⁵.

Dentro de la planificación de usuarios se tiene el concepto de adaptación de enlace, el mismo que para una potencia de transmisión conocida optimiza la capacidad y la cobertura, entonces el transmisor adapta la tasa de cada usuario a las variaciones de calidad de la señal que recibe, es decir, de esta manera se produce la modulación y codificación adaptativas AMC ⁶.

Dentro de la modulación, se tienen los esquemas QPSK, 16-QAM o 64-QAM, en los cuales los bits por símbolo dependen del esquema seleccionado, QPSK 2 bits por símbolo, 16-QAM 4 bits por símbolo y 64-QAM 6 bits por símbolo, generando 4, 16 ó 64 niveles respectivamente. La modulación QPSK tiene tasas binarias bajas pero presenta mejor resistencia a la interferencia y al ruido; mientras que para 64-QAM, las tasas binarias son altas pero la resistencia al ruido y a la interferencia son bajas, es decir necesita una alta relación de SINR ⁷.

⁵MCS: *Modulation and Coding Scheme*.

⁶AMC: *Adaptive Modulation and Coding*.

⁷SINR: *Signal to Interference and Noise Ratio*, señal/(ruido+interferencia)

Dentro de la codificación se selecciona la tasa del código de canal de corrección de errores, el mismo que es igual a la división del número de bits de entrada para el número de bits de salida; en donde para una menor tasa del código es mejor la corrección de errores, pero la tasa binaria de la señal es baja.

La radio base (eNB), realiza la AMC, seleccionando la modulación y codificación; esta selección es basada en el parámetro CQI que le envía el equipo de usuario UE.

Para enviar el CQI, el UE se basa en las señales de referencia RS que le envía el eNB. De la medición de las RS se calcula la relación señal/ruido del canal descendente, y notifica al eNB mediante el canal PUSCH, con un CQI que garantice una tasa de error de bloque (BLER) menor al 10 %. Dentro del canal PUSCH se envía el CQI en conjunto con el mejor MCS seleccionado.

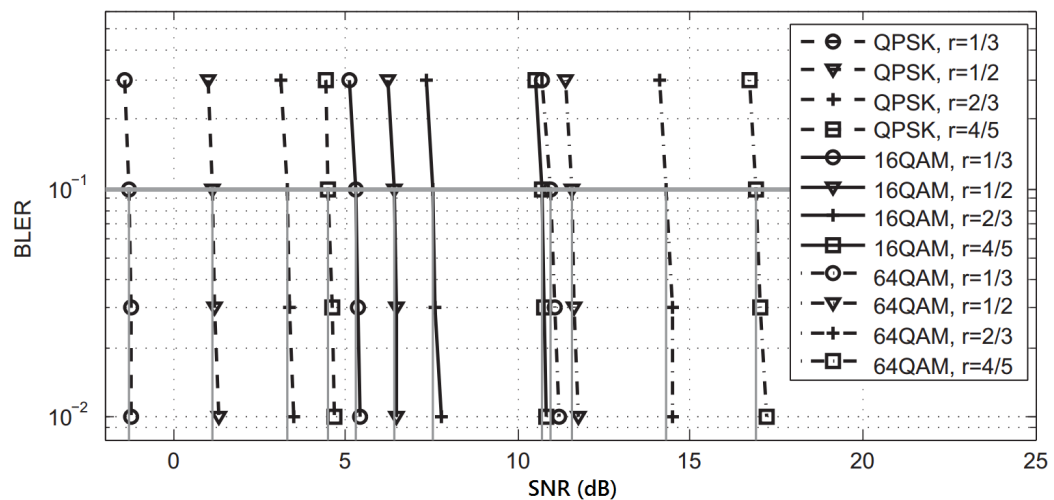


Figura 2.30: BLER vs relación señal/ruido (SNR) para diferentes MCS. [14]

En la Figura 2.30 se muestran los valores típicos del BLER de acuerdo a la relación señal/ruido para diferentes modulaciones y tasas de codificación.

Como ejemplo se observa en la Figura 2.30 una línea horizontal en el BLER=10 %, y para un valor de SNR=13 dB, se tiene que los valores MCS corresponden a una modulación 64-QAM con una tasa de codificación igual a 1/2.

En la Tabla 2.5 se observa los esquemas de modulación y tasas de código que pueden ser señalados mediante un valor CQI, en donde la eficiencia es la información de bits por símbolo, y se calcula multiplicando la tasa de código por el logaritmo en base dos de M , en donde $M = 2^n$, siendo n el número de bits del esquema de modulación. Para calcular SNR, se aplica la fórmula de Shannon, $2^{\text{eficiencia}}$ menos 1.

Tabla 2.5: Valores de CQI. [14]

INDICE CQI	MODULACIÓN	TASA DE CÓDIGO	EFICIENCIA	SNR (dB)
0	Fuera del rango			
1	QPSK	0.076	0.1523	-9.53
2	QPSK	0.12	0.2344	-7.53
3	QPSK	0.19	0.377	-5.25
4	QPSK	0.3	0.6016	-2.86
5	QPSK	0.44	0.877	-0.78
6	QPSK	0.59	1.1758	1.00
7	16-QAM	0.37	1.4766	2.51
8	16-QAM	0.48	1.9141	4.42
9	16-QAM	0.6	2.1063	5.19
10	64-QAM	0.45	2.7305	7.51
11	64-QAM	0.55	3.3223	9.54
12	64-QAM	0.65	3.9023	11.45
13	64-QAM	0.75	4.5234	13.42
14	64-QAM	0.85	5.1152	15.27
15	64-QAM	0.93	5.5547	16.63

Una vez que se conoce el valor del CQI, la eNB puede seleccionar que MCS se utilizan para poder decodificar la información del canal descendente. Es necesario una notificación periódica del CQI, para obtener una mejor adaptación del canal.

El equipo de usuario, transmite el CQI por el canal PUCCH, el eNB responde al UE mediante el canal de control de enlace descendente PDCCH.

En la sección 2.3.4 se describen brevemente las características que presentan cada uno de los procesos de la capa física de la red 4G LTE.

2.3.4. Codificación

La codificación se realiza en la capa física con el objetivo de corregir errores como la pérdida de información por desvanecimiento de canal. Para corregir estas pérdidas de información, existen dos métodos que se utilizan en LTE:

- **Solicitud de Respuesta Automática, ARQ (*Automatic Repeat Request*)**: Si se detecta un error, el receptor solicita la retransmisión de los paquetes de datos, hasta que se cumplan una de las siguientes condiciones: recibir los paquetes sin errores o se alcanza un número máximo de retransmisiones.
- **Corrección de errores hacia adelante, FEC (*Forward Error Correction*)** : Se añaden bits de redundancia a los bits de datos, lo cual permite la corrección de errores. En este caso se utilizan dos tipos de códigos: códigos de bloque y códigos convolucionales.

En los códigos de bloque, los bloques o vectores de datos de entrada, son multiplicados por una matriz generadora, que nos da como resultado un vector de palabra de código. En LTE, el código de bloque utilizado es el CRC (*Cyclic Redundancy Check*), el cual es un código de bloque lineal cíclico.

Por otro lado los códigos convolucionales tienen memoria, es decir que el bit de la salida codificada depende no sólo de los siguientes bits, sino también de los m bits anteriores, siendo m el número de bits guardados en el registro en el codificador convolucional.

En los canales de control de LTE se utilizan códigos convolucionales *Tail Biting* y en los canales de datos, es decir los que llevan la información, se utilizan códigos turbo, debido a que estos poseen mayor eficiencia que los códigos convolucionales.

Codificador Convolucional

Los códigos convolucionales constituyen el esquema de codificación de canal más básico que se implementa en los sistemas móviles de nueva generación.

Un codificador convolucional $C(k, n, m)$ consiste de un registro de desplazamiento con m etapas, que al pasar por un operador XOR se obtiene el bit de salida. Si se tiene k bits de información de entrada, se producen n bits de salida, de esta manera la tasa del código viene dada por $R = k/n$.

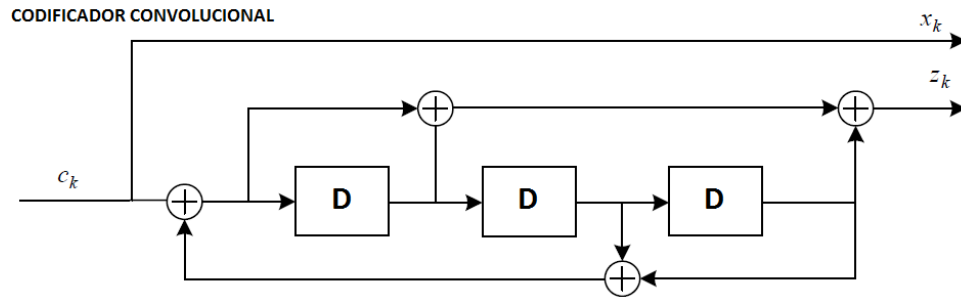


Figura 2.31: Codificador convolucional $k = 1$, $n = 3$, $m = 3$ y tasa $R = 1/3$. [3]

Turbo Código

El codificador turbo en LTE está formado en el transmisor por dos codificadores convolucionales con dos codificadores de Estado-8, trabajando en paralelo y separados por un entrelazador llamado concatenación paralela de códigos convolucionales (PCCC). Cada bloque de datos se codifica dos veces: primero se alimenta el codificador convolucional con la secuencia de bits original y posteriormente se alterna el orden de la misma con el entrelazador.

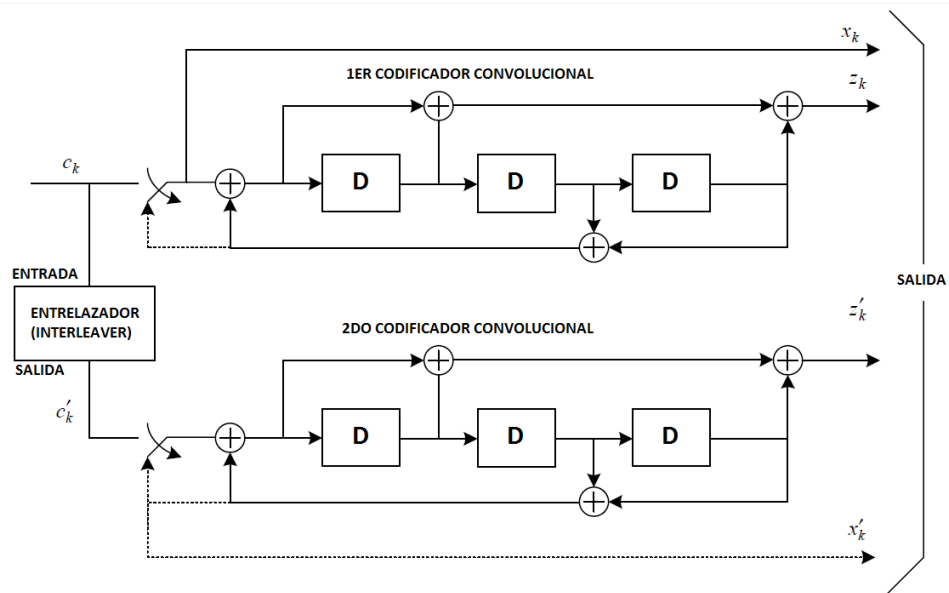


Figura 2.32: Estructura de codificador turbo para LTE. [3]

Los turbo códigos permiten mejorar la capacidad de canal, logrando un desempeño cercano a los límites teóricos de la capacidad. Otro aspecto en el que se mejora es en la eficiencia en el costo del sistema, ya que el usuario podrá enviar la misma cantidad de información empleando la mitad del ancho de banda. Los códigos turbo se emplean para codificar información de usuario, pues son más eficientes que los códigos convolucionales.

Decodificación de un Código Convolucional

El método utilizado para la decodificación de las secuencias codificadas mediante convolución, es el algoritmo de Viterbi, debido a su eficiencia, en donde el receptor debe conocer los estados inicial y final, para lo cual se puede utilizar diferentes métodos.

El algoritmo de Viterbi consiste en encontrar el camino óptimo basado en la métrica, es decir, compara cada bit de la palabra de código con todos los caminos posibles de Trellis, y elimina los caminos menos probables, esta comparación se realiza mediante el cálculo de la distancia de Hamming.⁸

Tabla 2.6: Estados de Trellis. [16]

ESTADO	BITS
A	000
B	001
C	010
D	011
E	100
F	101
G	110
H	111

La decodificación se basa en los estados en los que está el convolucionador, Tabla 2.6, y según el cálculo de la métrica se determina cual fue el bit de entrada, de esta manera se conoce el mensaje original. [16]

⁸Distancia de Hamming: Posiciones en las cuales son diferentes dos palabras de código

Si se tiene el estado $A = 000$, estado inicial, el bit de entrada puede ser 1 o 0, dependiendo de esto, se obtiene la salida del convolucional, es decir, para la entrada 0, la salida puede ser 000 o 001, dependiendo si el *interleaver* cambia el bit de entrada y el estado del convolucionador no cambia. Para el caso que la entrada sea 1, la salida puede ser 111 o 110 si el *interleaver* cambia el bit de entrada, y el estado del convolucionador cambia al E . Si el mensaje que llega es 001, se compara con cada salida, y se obtiene su métrica, para este ejemplo la métrica con la salida 000 es 1, la métrica con la salida 001 es 0, la métrica con la salida 111 es 2, y la métrica con la salida 110 es 3, entonces se dice que el mensaje original es un 0, Figura 2.33.

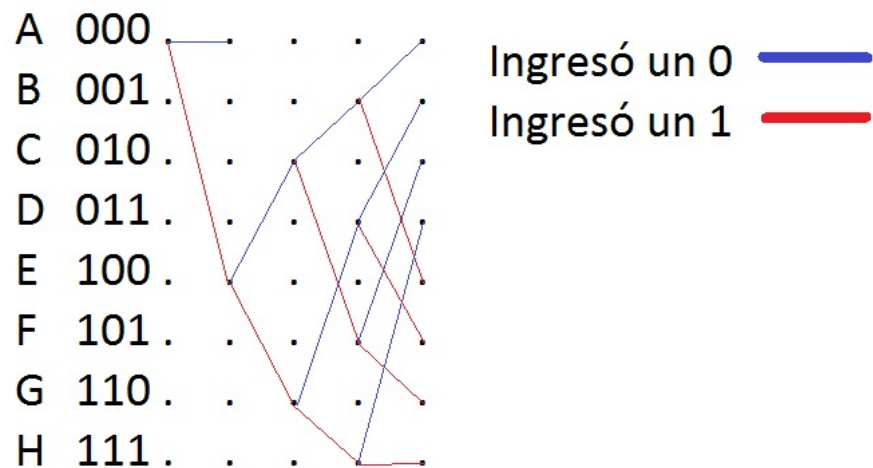


Figura 2.33: Diagrama de Trellis. [4]

Precodificación

Otro modo de mejorar el comportamiento de un sistema OFDM frente a la variabilidad que presentan los canales móviles es el empleo de estrategias de pre codificación. Su uso mejora la eficiencia de los sistemas de comunicación. La base de pre-codificación consiste en pre-conformar la información a transmitir en cada portadora con el objetivo de mejorar algún aspecto de la transmisión, como puede ser obtener una BER más baja en el receptor sin necesidad de aumentar la información redundante enviada.

Pre-codificación es uno de los enfoques más eficaces adoptados por el estándar LTE para superar el problema de la deficiencia de rango. Resulta de mucha utilidad emplear técnicas de pre-codificación que no necesiten disponer de ninguna información a priori del canal en el transmisor.

El objetivo principal de los pre-codificadores independientes del canal es evitar la pérdida completa de la información contenida en las portadoras que sufren desvanecimientos profundos. Una solución para tratar de resolver esta pérdida consiste en redistribuir la información contenida en varios símbolos QPSK/QAM entre varias portadoras, consiguiéndose aumentar la diversidad de la señal transmitida sin comprometer la eficiencia.

2.3.5. Multiplexación por División de Frecuencias

La Multiplexación por división de Frecuencias Ortogonales (OFDM), consiste en dividir el flujo de datos original en sub-informaciones paralelas en el dominio de la frecuencia, y transmitir éstas en distintas frecuencias llamadas sub-portadoras, permitiendo de esta manera reducir la interferencia entre símbolos al transmitir a altas velocidades de datos y aumentar la eficiencia espectral. Según la norma 3GPP, la interferencia entre símbolos es insignificante, puesto que LTE puede utilizar hasta un máximo de 1200 subportadoras.

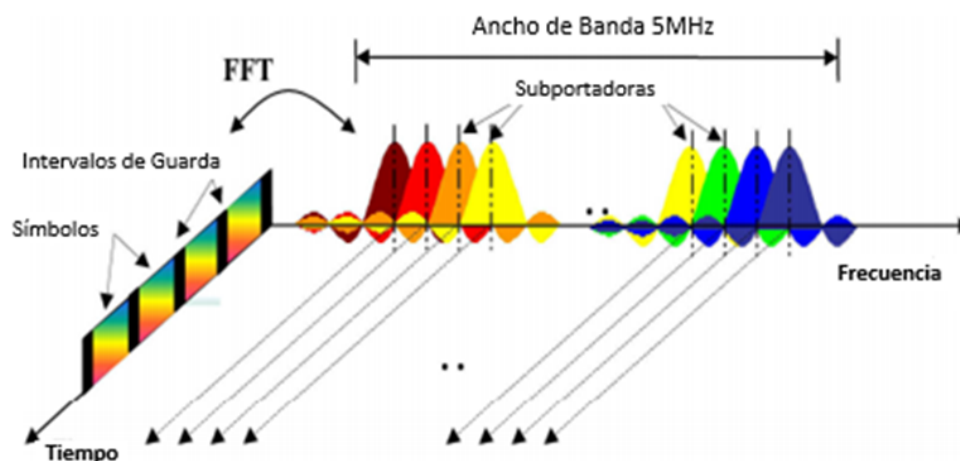


Figura 2.34: Representación de una señal OFDM. [3]

Se observa en la Figura 2.34, una señal ODFM con un ancho de banda de 5MHz, el mismo que se divide en varias subportadoras, mediante las cuales se transmiten los datos modulados. El intervalo de guarda, o prefijo cíclico se agrega a cada símbolo, el cual permite disminuir la interferencia intersímbolo, y cuyo resultado es la diferencia de tiempo entre símbolos receptados de la primera señal y la última señal multicamino, este intervalo de guarda viene dado en μs . La separación entre subportadoras para LTE es de 15KHz. [3]

El sistema de comunicaciones LTE 4G, utiliza para el enlace descendente OFDMA (*Orthogonal Frequency Division Multiple Access*), pues permite el acceso simultáneo de múltiples usuarios al sistema asignándoles a cada uno recursos en tiempo y frecuencia.

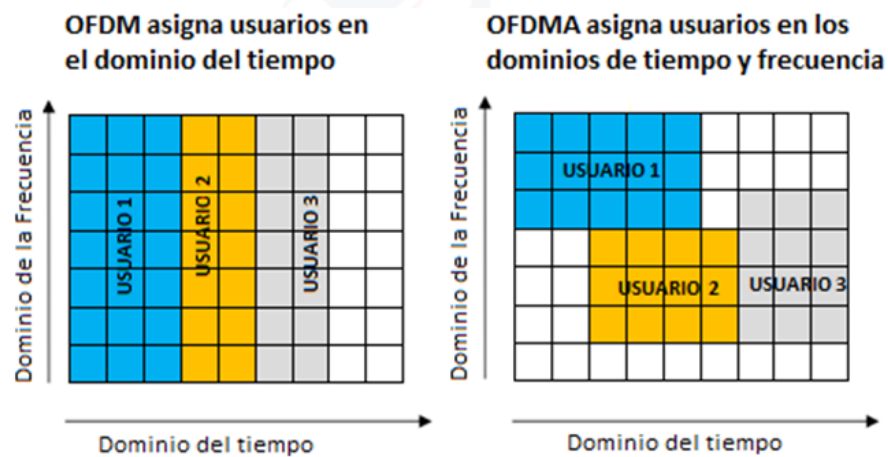


Figura 2.35: Asignación de en tiempo y frecuencia OFDMA. [3]

Para el enlace ascendente, LTE utiliza SC-FDMA (*Single Carrier Frequency Division Multiple Access*), ya que la relación entre la potencia pico instantánea y la potencia media de la señal a transmitir es menor, sin alterar la flexibilidad al momento de asignar las subportadoras de manera igual que OFDMA. La diferencia entre SC-FDMA y OFDMA, es que en la primera cada subportadora contiene información de todos los símbolos transmitidos, mientras en la segunda cada subportadora contiene información de un símbolo específico. Figura 2.36.

Cabe mencionar que SC-FDMA en esquema se diferencia por la presencia de un bloque DFT (*Discrete Fourier Transform*), cuya función es mezclar los símbolos antes de mapearlos sobre las subportadoras, y de esta manera distribuir la señal sobre el dominio de la frecuencia reduciendo las variaciones de potencia.

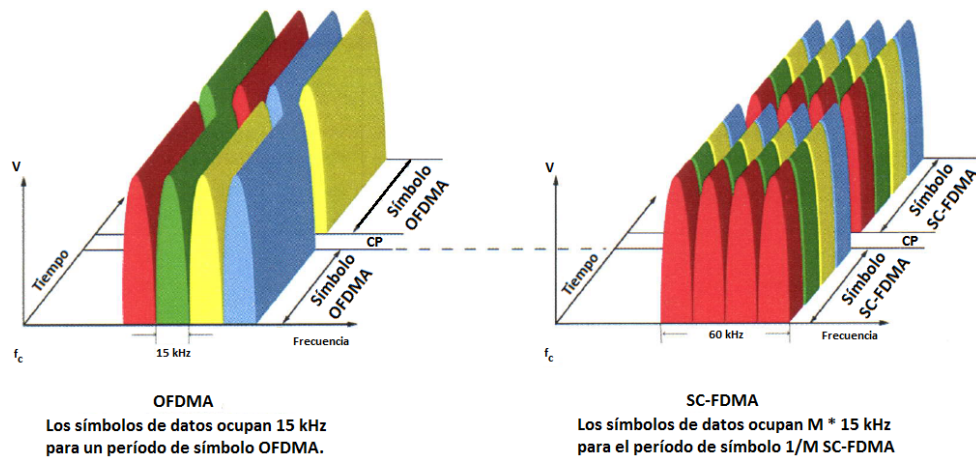


Figura 2.36: Diferencias entre subportadoras. [2]

2.3.6. Modulación

En cuanto a modulación, entre las principales características de LTE se encuentra el uso de la modulación de forma adaptativa, mediante modulaciones como: QPSK, 16QAM y 64QAM. El uso de modulación adaptativa permite que el sistema inalámbrico pueda escoger el orden de modulación en función de las condiciones del canal. Es decir Los diferentes órdenes de modulación permiten enviar más bits por símbolo y, por tanto, alcanzar un mayor *throughput*⁹ y eficiencia espectral. [8]

A pesar de ello, utilizar técnicas de modulación como 64-QAM, implica que sea necesaria una mayor relación señal a ruido (SNR) para evitar las interferencias y mantener una tasa de error de bit moderada.

Para el enlace ascendente, los esquemas de modulación posibles para el canal PUSCH, según la 3GPP [1], son: QPSK, 16 QAM Y 64 QAM. Y para el enlace descendente, los esquemas de modulación posibles, tanto para el canal PDSCH como para el PMCH según la 3GPP [1] son: QPSK, 16 QAM, 64 QAM y 256 QAM.

⁹ *Throughput*: Tasa promedio de éxito de entrega de la información sobre un canal de comunicación.

2.3.7. Modulación QPSK - Modulación de Fase en Cuadratura

La modulación digital de fase o PSK (*Phase Shift Keying*), es aquella en donde se codifica por cambio de fase, es decir se alterna la fase para transmitir la información, para cada valor de la señal digital, se asigna un desfase de la señal analógica (0 se modifica la fase, 1 no se modifica la fase), empleando un método de sincronismo para la transmisión y recepción de datos. Las fases que se utilizan están dadas por $2^n = M$ donde $2 \leq n \leq 16$.

La modulación QPSK (Quaternaria PSK) o modulación de fase en cuadratura se presentan 4 fases equidistantes, que pueden ser representadas por los dígitos 00, 01, 10, 11, cada fase transmite dos bits.

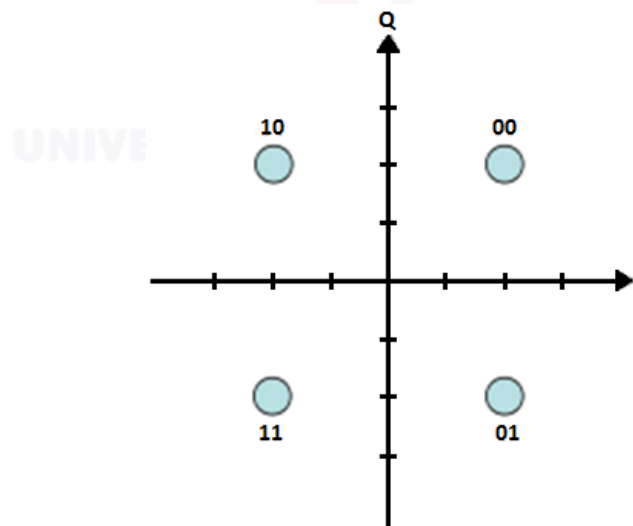


Figura 2.37: Constelación de modulación QPSK. [5]

2.3.8. Modulación QAM- Modulación de Amplitud en Cuadratura

La modulación QAM es una modulación digital en la cual el mensaje se encuentra tanto en la amplitud como en la fase de la señal transmitida, es decir nos permite transportar dos mensajes de forma independiente.

Esta modulación se caracteriza por permitirnos enviar dos mensajes independientes por un único camino, esto se realiza mediante la formación de dos canales ortogonales en el mismo ancho de banda, es decir, se modula una misma portadora, desfasada 90 grados entre los dos mensajes. Al implementar esta modulación permite el uso eficiente del ancho de banda.

La modulación QAM tiene como entrada un flujo de datos binarios, que se divide en grupos de bits como se requieran y así generar N estados de modulación, es por esto que existen N-QAM modulaciones, como son 16 QAM, 64 QAM, etc. En N-QAM, cada grupo de m -bits genera $2^m = N$ estados de modulación.

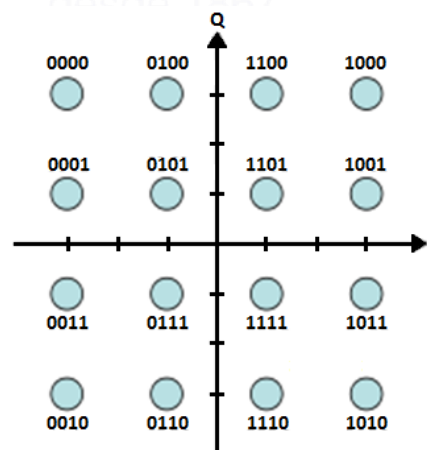


Figura 2.38: Constelación de modulación 16-QAM. [5]

El objetivo de aumentar el orden en una modulación QAM es el de transmitir mayor cantidad de bits por símbolo. Sin embargo, esto puede hacer que estos bits sean más susceptibles al ruido y la distorsión, y esto nos da una tasa de error de bits más alta. La 3GPP, define el tipo de modulación dependiendo del canal físico.

2.3.9. Técnicas Multi-Antena (MIMO)

Las técnicas MIMO (*Multiple Input Multiple Output*), son esenciales en LTE, éstas incluyen multiplexación espacial que consiste en enviar señales desde dos o más antenas diferentes con diferentes corrientes de datos, y el receptor, mediante procesamiento de señales separar las corrientes de datos. Ésto permite obtener altas velocidades de transmisión de datos y un eficiente uso del espectro radio-eléctrico, dependiendo del número de antenas que se utilicen. [7]

En LTE, las configuraciones soportadas según el 3GPP, son 2x2, que consiste en dos antenas en el UE y dos antenas en la estación base, y la configuración 4x4. Las mismas que idealmente permiten una velocidad de bajada de 150 Mbps en el caso de 2x2 MIMO, y de 300Mbps en el 4x4 MIMO, con un ancho de banda de 20MHz. [7]

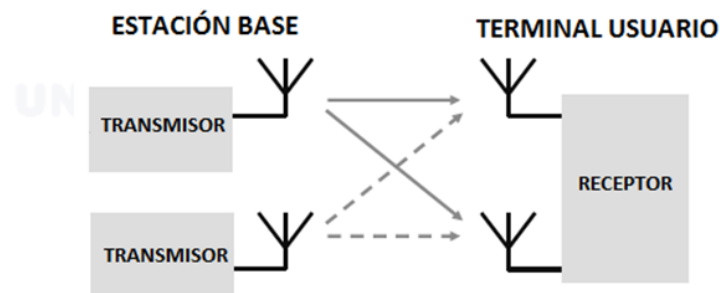


Figura 2.39: Técnica MIMO configuración 2x2. [3]

Los flujos de datos son modulados o codificados para su transmisión, es decir estos se multiplican por una matriz de un *codebook* predefinido, y seleccionado por el UE y conocido estación base, esta matriz es elegida de acuerdo a la estimación del canal.

Para la transmisión de datos, LTE soporta FDD (*Frequency Division Duplex*) y TDD (*Time Division Duplex*), dependiendo de la banda de frecuencia que se utilice:

- En FDD: la ENodeB y el UE, transmiten y reciben de manera simultánea, empleando distintas portadoras y de esta manera las velocidades de subida y bajada son aproximadamente iguales.

- En TDD: Transmiten y reciben en tiempos diferentes, pero en la misma frecuencia, es decir de manera asimétrica, la velocidad de bajada es mayor a la de subida.

La configuración multiantenas, permite aumentar la velocidad de datos, configurando múltiples flujos de datos, dependiendo del número de antenas en el emisor y en el receptor.

Existen varios tipos de configuración multi-antena, dependiendo de las antenas que se utilicen en el emisor y en el receptor. La configuración SISO, emplea una antena en cada extremo del enlace; la configuración SIMO, utiliza una antena en el transmisor y varias antenas en el receptor. La configuración MISO es el inverso de SIMO, múltiples antenas en el transmisor y una sola antena en el receptor. La configuración MIMO, emplea múltiples antenas en los extremos del enlace.

El enlace descendente permite varias configuraciones, una es la transmisión por una sola antena empleando una señal de referencia. La MIMO multiusuario en la cual se asigna a varios UE, recursos de tiempo y frecuencia.

El enlace ascendente permite dos configuraciones, la transmisión por una sola antena, y la pre-codificación con multiplexación adaptable, que permite adaptar el número de capas entre uno y cuatro.

Capítulo 3

Procesos de la Capa Física según la 3GPP

A continuación se detallan los procesos que abarca la normativa 3GPP [1], para los canales de transporte y control de información de LTE.

3.1. Codificación de Canal, Multiplexación y Entrelazado

Los flujos de datos y control de la capa MAC se codifican/decodifican para ofrecer servicios de transporte y de control a través del enlace de transmisión. El esquema del canal de codificación consta de procesos como la detección y corrección de errores, adaptación de velocidad, entrelazado; y un canal de transporte y control, para la asignación de información.

3.1.1. Código de bloque CRC

El código de comprobación de redundancia cíclica (CRC, *Cyclic Redundancy Check*) se caracteriza por la detección de errores ante algún cambio de datos. Los bloques de datos que entran en el sistema reciben un valor de comprobación, el cual se basa en el resto de una división polinómica; y en el proceso inverso se realiza el mismo cálculo, y se comprueba que los valores de comprobación estén correctos, caso contrario se corrigen los errores obtenidos.

Para crear los bits de paridad se aplica uno de los siguientes polinomios generadores cíclicos, donde L es el número de bits de paridad.

$$g_{CRC24A}(D) = D^{24} + D^{23} + D^{18} + D^{17} + D^{14} + D^{11} + D^{10} + D^7 + D^6 + D^5 + D^4 + D^3 + D + 1 \text{ para } L=24 \quad (3.1)$$

$$g_{CRC24B}(D) = [D^{24} + D^{23} + D^6 + D^5 + D + 1] \text{ para } L=24 \quad (3.2)$$

$$g_{CRC16}(D) = [D^{16} + D^{12} + D^5 + 1] \text{ para } L=16 \quad (3.3)$$

$$g_{CRC8}(D) = [D^8 + D^7 + D^4 + D^3 + D + 1] \text{ para } L=8 \quad (3.4)$$

Para la red 4G LTE, el enlace descendente utiliza los polinomios generadores 3.1 y 3.2; los mismos que tienen 24 bits de paridad; el primero (3.1) se centra en el bloque de transporte y el segundo (3.2) se centra en el bloque de código, que es la división de un bloque de transporte cuando sobrepasa el tamaño de 6144 bits que es el límite superior.

3.1.2. Código de segmentación en bloques y acoplamiento CRC

Para una secuencia de entrada dada por $b_0, b_1, b_2, \dots, b_{B-1}$ en donde $B > 0$, el tamaño del bloque para una longitud (L) de 24 bits es 6144 bits. en donde si el tamaño de la secuencia de entrada es $B > 40$, se añaden bits de relleno al principio del primer bloque, en donde estos bits de relleno adquieren un valor nulo en la entrada del codificador convolucional.

3.1.3. Codificación de Canal LTE

En la red 4G LTE, se utilizan dos esquemas de codificación de canal, dependiendo del canal de transporte o control que se va utilizar:

- Códigos convolucionales *Tail biting*.
- Turbo código.

Las Tablas 3.1 y 3.2, presentan los esquemas de codificación como su tasa de codificación dependiendo de los canales de transporte y control que se empleen.

Tabla 3.1: Esquema y velocidad de codificación para el canal de transporte. [1]

Tr-CH (CANAL DE TRANSPORTE)	ESQUEMA DE CODIFICACIÓN.	TASA DE CODIFICACIÓN
UL-SCH	TURBO CÓDIGO	1/3
DL-SCH		
PCH		
MHC		
BCH	CÓDIGO CONVOLUCIONAL <i>TAIL BITING</i>	1/3

Tabla 3.2: Esquema y velocidad de codificación para el canal de control. [1]

CONTROL DE INFORMACIÓN.	ESQUEMA DE CODIFICACIÓN.	TASA DE CODIFICACIÓN.
DCI	CÓDIGO CONVOLUCIONAL <i>TAIL BITING</i>	1/3
CFI	CÓDIGO DE BLOQUE	1/16
HI	CÓDIGO DE REPETICIÓN	1/3
UCI	CÓDIGO DE BLOQUE	VARIABLE
	CÓDIGO CONVOLUCIONAL <i>TAIL BITING</i>	1/3

3.1.4. Código Convolutacional Tail Biting

Estos códigos son utilizados para codificar información de control, puesto que el algoritmo de decodificación es más sencillo. Esta codificación viene definida con una longitud de 7 bits y una tasa de codificación de 1/3.

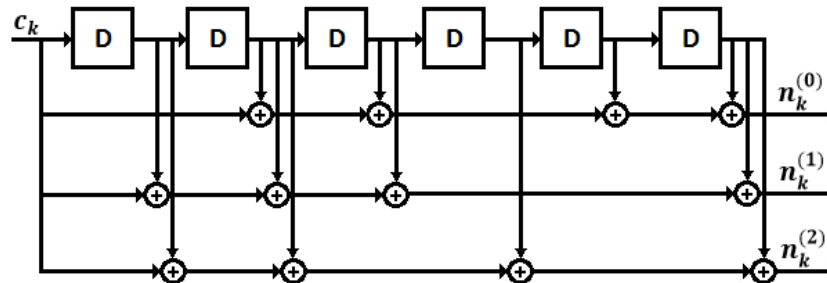


Figura 3.1: Codificador convolutacional con $k = 1$, $n = 3$, $m = 6$ tasa $R = 1/3$. [1]

3.1.5. Turbo Código.

- **Codificador turbo.**

El codificador turbo que se emplea en LTE, se representa en la Figura 3.2, el mismo que está compuesto por dos codificadores convolucionales, y un entrelazador (*interleaver*).

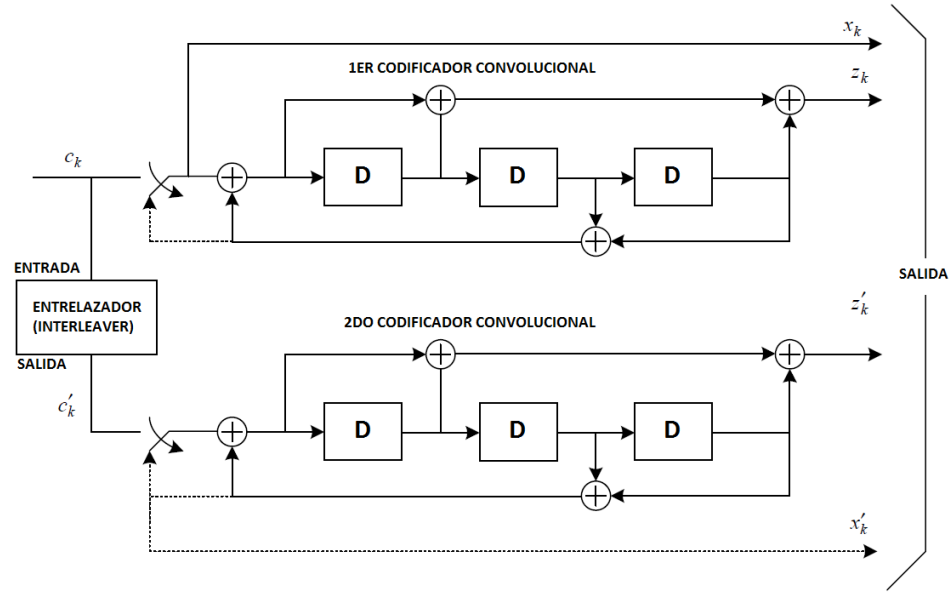


Figura 3.2: Estructura de codificador turbo con tasa 1/3. [3]

La función de transferencia que presenta este codificador es:

$$G(D) = \left[1, \frac{g_1(D)}{g_0(D)} \right] \quad (3.5)$$

En donde:

$$g_0(D) = 1 + D^2 + D^3 \quad (3.6)$$

$$g_1(D) = 1 + D + D^3 \quad (3.7)$$

La salida esta conformada por x_k, z'_k, z_k , para $k = 0, 1, 2, \dots, k-1$, en donde el bit x_k se le llama bit sistemático, y a los bits z_k, z'_k son los bit de paridad, los últimos bits del turbo código, llamados bits de Trellis ocupan el bit de salida x'_K , mediante la siguiente estructura:

$$d_K^{(0)} = x_K, d_K^{(1)} = z_K, d_K^{(2)} = x_{k+1}$$

$$d_{K+1}^{(0)} = z_{K+1}, d_{K+1}^{(1)} = x_{K+2}, d_{K+1}^{(2)} = z_{k+2}$$

$$d_{K+2}^{(0)} = x'_K, d_{K+2}^{(1)} = z'_K, d_{K+2}^{(2)} = x'_{K+1}$$

$$d_{K+3}^{(0)} = z'_{K+1}, d_{K+3}^{(1)} = x'_{K+2}, d_{K+3}^{(2)} = z'_{K+2}$$

El estado inicial de los codificadores convolucionales es cero.

- **Entrelazador Interno (*Interleaver*)**

En la Figura 3.2 los bits de entrada al segundo codificador convolucional pasan por un entrelazador de bits, cuya salida esta representada por la expresión:

$$c'_i = c_{\Pi(i)}, \text{ para } i=0,1,2,\dots,(k-1) \quad (3.8)$$

En donde la relación entre la entrada $c_{\Pi(i)}$ y los índices de salida cumple la siguiente ecuación.

$$\Pi(i) = (f_1 i + f_2 i^2) \bmod K \quad (3.9)$$

Para determinar el valor de f_1 y f_2 se considera el tamaño del bloque K , y se presenta en la Tabla 3.3.

La función $(f_1 i + f_2 i^2) \bmod K$, se describe como el residuo de la división entre $(f_1 i + f_2 i^2)$ y K , es decir, si $f_1 = 3$, $f_2 = 10$ y $K = 40$ entonces para $\Pi(2)$ el resultado sería 6.

UNIVERSIDAD DE CUENCA
desde 1867



CAPÍTULO 3. PROCESOS DE LA CAPA FÍSICA SEGÚN LA 3GPP

Tabla 3.3: Valores de f_1 y f_2 de acuerdo al tamaño de bloque K. [1]

i	K	f_1	f_2	i	K	f_1	f_2	i	K	f_1	f_2	i	K	f_1	f_2
1	40	3	10	48	416	25	52	95	1120	67	140	142	3200	111	240
2	48	7	12	49	424	51	106	96	1152	35	72	143	3264	443	204
3	56	19	42	50	432	47	72	97	1184	19	74	144	3320	51	104
4	64	7	16	51	440	91	110	98	1216	39	76	145	3392	51	212
5	72	7	18	52	448	29	168	99	1248	19	78	146	3456	451	192
6	80	11	20	53	456	29	14	100	1280	199	240	147	3520	257	220
7	88	5	22	54	464	247	58	101	1312	21	82	148	3584	57	336
8	96	11	24	55	472	29	118	102	1344	211	252	149	3648	313	228
9	104	7	26	56	480	89	180	103	1376	21	86	150	3712	271	232
10	112	41	84	57	488	91	122	104	1408	43	88	151	3776	179	236
11	120	103	90	58	496	157	62	105	1440	149	60	152	3840	331	120
12	128	15	32	59	504	55	84	106	1472	45	92	153	3904	363	244
13	136	9	34	60	512	31	64	107	1504	49	846	154	3968	375	248
14	144	17	108	61	528	17	66	108	1536	71	48	155	4032	127	168
15	152	9	38	62	544	35	68	109	1568	13	28	156	4096	31	64
16	160	21	120	63	560	227	420	110	1600	17	80	157	4160	33	130
17	168	101	84	64	576	65	96	111	1632	25	102	158	4224	43	264
18	176	21	44	65	592	19	74	112	1664	183	104	159	4288	33	134
19	184	57	46	66	608	37	76	113	1696	55	954	160	4352	477	408
20	192	23	48	67	624	41	234	114	1728	127	96	161	4416	35	138
21	200	13	50	68	640	39	80	115	1760	27	110	162	4480	233	280
22	208	27	52	69	656	185	82	116	1792	29	112	163	4544	357	142
23	216	11	36	70	672	43	252	117	1824	29	114	164	4608	357	480
24	224	27	56	71	688	21	86	118	1856	57	116	165	4672	37	146
25	232	85	58	72	704	155	44	119	1888	45	354	166	4736	71	444
26	240	29	60	73	720	79	120	120	1920	31	120	167	4800	71	120
27	248	33	62	74	736	139	92	121	1952	59	60	168	4864	37	152
28	256	15	32	75	752	23	94	122	1984	185	124	169	4928	39	462
29	264	17	98	76	768	217	48	123	2016	113	420	170	4992	127	234
30	272	33	68	77	784	25	98	124	2048	31	64	171	5056	39	158
31	280	109	20	78	800	17	80	125	2112	17	66	172	5120	39	80
32	288	19	36	79	816	127	102	126	2176	171	136	173	5184	31	96
33	296	19	74	80	832	25	52	127	2240	209	420	174	5248	113	902
34	304	37	76	81	848	239	106	128	2304	253	216	175	5312	41	166
35	312	19	78	82	864	17	48	129	2368	367	444	176	5376	251	336
36	320	21	12	83	880	137	110	130	2432	265	456	177	5440	43	170
37	328	21	82	84	896	215	112	131	2496	181	468	178	5504	21	86
38	336	115	84	85	912	29	114	132	2560	39	80	179	5568	43	174
39	344	193	86	86	928	15	58	133	2624	27	164	180	5632	45	176
40	352	21	44	87	944	147	118	134	2688	127	504	181	5696	45	178
41	360	133	90	88	960	29	60	135	2752	143	172	182	5760	161	120
42	368	81	46	89	976	59	122	136	2816	43	88	183	5824	89	182
43	376	45	94	90	992	65	124	137	2880	29	300	184	5888	323	184
44	384	23	48	91	1008	55	84	138	2944	45	92	185	5952	47	186
45	392	243	98	92	1024	31	64	139	3008	57	188	186	6016	23	94
46	400	151	40	93	1056	17	66	140	3072	47	96	187	6080	47	190
47	408	155	102	94	1088	171	204	141	3136	13	28	188	6144	263	480

3.2. Adaptación de Velocidad

El proceso de *Rate Matching*, o adaptación de velocidad, cumple la función de hacer coincidir el número de bits del bloque de transporte, con el número de bits que se puede transmitir en la modulación seleccionada. Este proceso consta de 3 partes, un intercalador por cada bit que se tiene a la salida del turbo código, la segunda parte es un proceso de selección de bits, y a tercera un *buffer* circular, como se puede observar en la Figura 3.3.

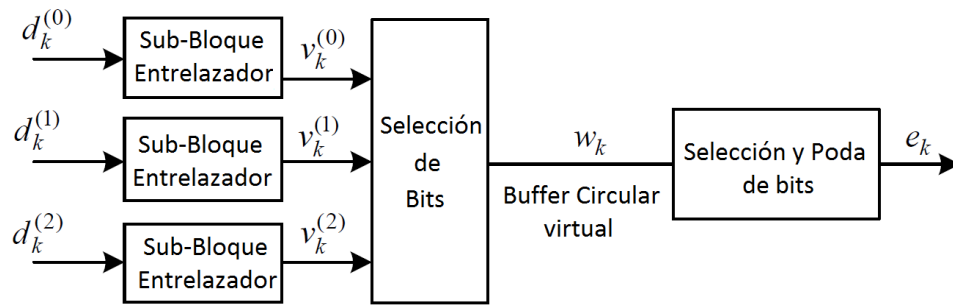


Figura 3.3: Procesos del *rate matching*. [1]

El bloque entrelazador tiene como entradas $d_k^{(0)}$, $d_k^{(1)}$ y $d_k^{(2)}$, que corresponden a los bits de salida del turbo código, es decir, $d_k^{(0)}$ es el bit sistemático, $d_k^{(1)}$ es el primer bit de paridad, y $d_k^{(2)}$ el segundo bit de paridad.

3.2.1. Sub-Bloque Entrelazador

Las entradas a cada sub-bloque entrelazador están dadas por $d_0^{(i)}$, $d_1^{(i)}$, $d_i^{(2)}$, ..., $d_{D-1}^{(i)}$, en donde i es el identificador de cada entrada (0 bit sistemático, 1 primer bit de paridad, y 2 segundo bit de paridad) y D es el número total del mensaje (Ecuación 3.10).

$$D = \text{Bits Mensaje} + 24 \text{ Bits CRC} + 4 \text{ Bits Trellis} + N \text{ and Asignadas} \quad (3.10)$$

La salida para cada una de las entradas está definida por las siguientes secuencias, en donde K_{Π} , se define mas adelante.

$$v_k^{(0)} = v_0^{(0)}, v_1^{(0)}, v_2^{(0)}, \dots, v_{K_{\Pi}-1}^{(0)}$$

$$v_k^{(1)} = v_0^{(1)}, v_1^{(1)}, v_2^{(1)}, \dots, v_{K_{\Pi}-1}^{(1)}$$

$$v_k^{(2)} = v_0^{(2)}, v_1^{(2)}, v_2^{(2)}, \dots, v_{K_{\Pi}-1}^{(2)}$$

La secuencia de bits de salida del entrelazador es una matriz cuyo número de columnas es igual a 32 ($C = 32$), numeradas desde 0 hasta 31. El número de filas de la salida, tiene que cumplir la relación $D \leq (C \times F)$, en donde F es el número de filas. Despejando F se obtiene el número de filas de la matriz de salida ($F = D/C$).

Si el número de filas por el número de columnas es mayor al número total del mensaje, $(C \times F) > D$, entonces se agregan N_D bits ficticios de manera que $y_K = NULL$ para $K = 0, 1, \dots, N_D - 1$ y $y_{N_D+K} = d_k^{(i)}$ para $K = 0, 1, \dots, D - 1$. Es decir, si D es mayor a $R \times C$, entonces se encuentra la diferencia, y esta diferencia son los bits ficticios que se agregan al principio de la matriz. La multiplicación entre el número de filas por el número de columnas se define como K_{Π} .

La matriz de bits de salida del entrelazador se va llenando según lo expuesto en la Ecuación 3.11.

$$\begin{bmatrix} y_0 & y_1 & y_2 & \dots & y_{C-1} \\ y_C & y_{C+1} & y_{C+2} & \dots & y_{2C-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{(F-1) \times C} & y_{(F-1) \times C+1} & y_{(F-1) \times C+2} & \dots & y_{(F \times C)-1} \end{bmatrix} \quad (3.11)$$

Para realizar la permutación entre columnas la norma 3GPP establece el siguiente patrón:

$$P(j) = P(0), P(1), \dots, P(C - 1) \quad (3.12)$$

$$P(j) = (0, 16, 8, 24, 4, 20, 12, 28, 2, 18, 10, 26, 6, 22, 14, 30, 1, 17, 9, 25, 5, 21, 13, 29, 3, 19, 11, 27, 7, 23, 15, 31) \quad (3.13)$$

Y la matriz permutada se forma a continuación:

$$\begin{bmatrix} y_{P(0)} & y_{P(1)} & y_{P(2)} & \dots & y_{P(C-1)} \\ y_{P(0)+C} & y_{P(1)+C} & y_{P(2)+C} & \dots & y_{P(C-1)+C} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{P(0)+(F-1)\times C} & y_{P(1)+(F-1)\times C} & y_{P(2)+(F-1)\times C} & \dots & y_{P(C-1)+(F-1)\times C} \end{bmatrix} \quad (3.14)$$

Esta matriz permutada es la salida del entrelazador, para $d_k^{(0)}$ y $d_k^{(1)}$, que esta dada por $v_0^{(i)}, v_1^{(i)}, v_2^{(i)}, \dots, v_{K_\Pi-1}^{(i)}$, en donde $y_{P(0)}$ corresponde a la salida $v_0^{(i)}$; $y_{P(0)+C}$ corresponde a la salida $v_1^{(i)}$, y así sucesivamente.

Para $d_k^{(2)}$ la salida del entrelazador esta dada por $v_0^{(2)}, v_1^{(2)}, v_2^{(2)}, \dots, v_{K_\Pi-1}^{(2)}$ donde $v_k^{(2)} = y_{\Pi(K)}$, y $\Pi(K)$ viene dado por la Ecuación 3.15, donde la función P , esta dada por la Ecuación 3.13.

$$\Pi(k) = \left(P \left(\left\lfloor \frac{k}{F} \right\rfloor \right) + C \times (k \bmod F) + 1 \right) \bmod K_\Pi \quad (3.15)$$

3.2.2. Buffer Circular y Selección de bits

La longitud del *buffer* circular esta dado por $K_w = 3K_\Pi$, y se genera mediante la combinación de las salidas $v_k^{(0)}, v_k^{(1)}, v_k^{(2)}$, siguiendo las secuencias:

$$\begin{pmatrix} w_k = v_k^{(0)} \\ w_{K_\Pi+k} = v_k^{(1)} \\ w_{2K_\Pi+k} = v_k^{(2)} \end{pmatrix} \quad k = 0, \dots, K_\Pi - 1$$

Para obtener la salida e_k se realiza un cálculo que depende del tamaño de bloque de código (N_{cb}), del tamaño del bloque de transporte (N_{IR}), los cuales estan dados por las Ecuaciones 3.16, 3.17 y 3.18.

$$N_{cb} = \min \left(\left\lfloor \frac{N_{IR}}{C} \right\rfloor, K_w \right) \text{ para DL-SCH} \quad (3.16)$$

$$N_{cb} = K_w \text{ para UL-SCH} \quad (3.17)$$

$$N_{IR} = \left\lfloor \frac{N_{soft}}{K_c \times K_{MIMO} \times \min(M_{DLHARQ}, M_{limit})} \right\rfloor \quad (3.18)$$

En donde N_{soft} , número total de bits del canal *soft*, depende de las características del UE, si el UE esta configurado en modo de transmisión 3, 4 o 9 y es de categoría-v1020, entonces $N_{soft} = 35982720$ y $K_C = 5$, si el UE, soporta no más de dos capas para el enlace descendente, $N_{soft} = 3654144$ y $K_C = 2$, para otros casos $K_C = 1$.

Para K_{MIMO} , puede tomar los valores de 1 o 2, en caso de que el UE este configurado para recibir mediante el canal PDSCH, en los modos de transmisión 3,4,8,9; $K_{MIMO} = 2$ para los demás casos $K_{MIMO} = 1$.

M_{DLHARQ} , es el número máximo de repeticiones HARQ, en donde para LTE $M_{DLHARQ} = 8$ para los enlaces ascendente y descendente. M_{limit} es una constante igual a 8.

Se define como E la longitud de la salida del *rate matching*, rv_{idx} como el índice de versión de redundancia y G es el número total de bits disponibles para la transmisión de un bloque de transporte, entonces:

$$G' = G / (N_L \times Q_m) \quad (3.19)$$

Donde Q_m depende de la modulación que se ocupe, es decir, para QPSK $Q_m = 2$, para 16 QAM $Q_m = 4$ y para 64 QAM $Q_m = 6$, y N_L es el número de capas de mapeo de un bloque de transporte, en caso de la diversidad de transmisión $N_L = 2$.

Se establece $\gamma = G' \bmod C$, donde C es el número de bloques de código. Entonces si $r \leq C - \gamma - 1$; se define $E = N_L \times Q_m \times \lfloor G'/C \rfloor$; caso contrario $E = N_L \times Q_m \times \lceil G'/C \rceil$.

Ahora para encontrar la salida e_k , depende de una constante k_0 , la misma que viene dada por:

$$k_0 = F \left(2 \left\lceil \frac{N_{cb}}{8F} \right\rceil rv_{idx} + 2 \right) \quad (3.20)$$

Entonces la salida e_k , se obtiene desde la posición k_0 de la salida w_k , es decir la posición $e_k(0) = k_0$ hasta $K_{\Pi} - 1$; y los últimos bits se definen por $0, 1, \dots, k_0 - 1$.

3.3. Aleatorización

En el procesamiento de enlace descendente, la aleatorización o *scrambling*, se realiza para que las transmisiones de diferentes celdas sean separadas antes de la decodificación; es decir se utilizan diferentes secuencias de *scrambling* en las celdas vecinas para asegurar que la interferencia sea aleatorizada. Siendo la secuencia única para cada celda, los generadores se basan en la identidad de la celda PHY, y se aplica en todos los canales de transporte y los canales de control.

El proceso de *scrambling* para el enlace descendente consiste en realizar una operación XOR, entre la salida del *rate matching* y una secuencia *gold* de longitud 31. Las secuencias *gold* vienen dadas por polinomios:

$$p_1(x) = x^{31} + x^3 + 1 \quad (3.21)$$

$$p_2(x) = x^{31} + x^3 + x^2 + x + 1 \quad (3.22)$$

El valor de inicialización de la primera secuencia, Ecuación 3.21 se especifica con una función de impulso unitario de longitud 31.

$$p_1(0) = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

El valor de inicialización para la segunda secuencia aleatoria depende de parámetros tales como la identidad de la celda, el número de palabras de código y el índice de subtrama. Finalmente, la multiplicación de nivel de bit se implementa como una operación OR-exclusivo entre los bits del *rate matching* y los bits de la secuencia *gold*.

El cálculo de las condiciones iniciales para p_2 , se realiza mediante la Ecuación 3.23, una vez obtenido el resultado, se transforma a número binario de longitud 31.

$$p_2(0) = RNTI \times 2^{14} + q \times 2^{13} + \lfloor ns/2 \rfloor \times 2^9 + IDCelda \quad (3.23)$$

En donde $RNTI$ es el ID del usuario; q es el número de palabras de código; $IDCelda$ es el ID de la celda en la cual se está trabajando.

En la Figura 3.4, se observa el flujo de la secuencia *gold*, en donde para evitar la redundancia, LTE realiza $2^n - 1$ iteraciones, y obtiene la secuencia *gold* con la cual se va a realizar la operacion XOR con la salida del *rate matching*.

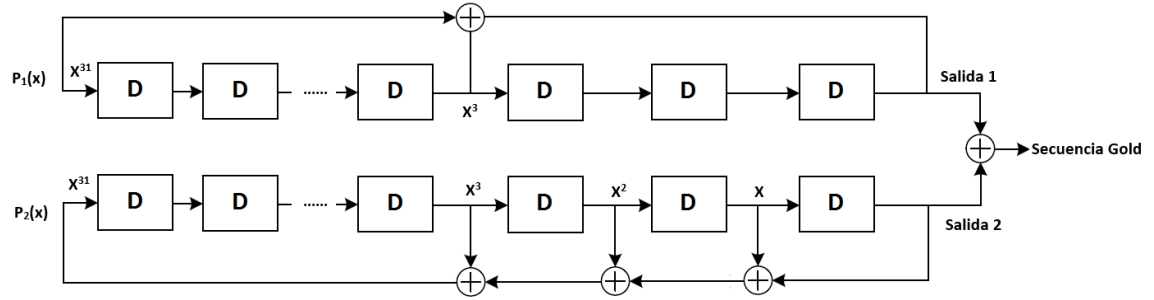


Figura 3.4: Diagrama secuencias gold. [4]

3.4. Modulaciones

La 3GPP establece el mapeo de las modulaciones para 4G LTE, el esquema de modulación BPSK se presenta en la Tabla 3.4; QPSK en la Tabla 3.5, el esquema de la modulación 16-QAM en la Tabla 3.6, y de la 64-QAM en la Tabla 3.7.

Tabla 3.4: Modulación BPSK. [1]

b(i)	I	Q
0	$1/\sqrt{2}$	$1/\sqrt{2}$
1	$-1/\sqrt{2}$	$-1/\sqrt{2}$

Tabla 3.5: Modulación QPSK. [1]

b(i)	I	Q
00	$1/\sqrt{2}$	$1/\sqrt{2}$
01	$1/\sqrt{2}$	$-1/\sqrt{2}$
10	$-1/\sqrt{2}$	$1/\sqrt{2}$
11	$-1/\sqrt{2}$	$-1/\sqrt{2}$

Tabla 3.6: Modulación 16-QAM. [1]

b(i)	I	Q
0000	$1/\sqrt{10}$	$1/\sqrt{10}$
0001	$1/\sqrt{10}$	$3/\sqrt{10}$
0010	$3/\sqrt{10}$	$1/\sqrt{10}$
0011	$3/\sqrt{10}$	$3/\sqrt{10}$
0100	$1/\sqrt{10}$	$-1/\sqrt{10}$
0101	$1/\sqrt{10}$	$-3/\sqrt{10}$
0110	$3/\sqrt{10}$	$-1/\sqrt{10}$
0111	$3/\sqrt{10}$	$-3/\sqrt{10}$
1000	$-1/\sqrt{10}$	$1/\sqrt{10}$
1001	$-1/\sqrt{10}$	$3/\sqrt{10}$
1010	$-3/\sqrt{10}$	$1/\sqrt{10}$
1011	$-3/\sqrt{10}$	$3/\sqrt{10}$
1100	$-1/\sqrt{10}$	$-1/\sqrt{10}$
1101	$-1/\sqrt{10}$	$-3/\sqrt{10}$
1110	$-3/\sqrt{10}$	$-1/\sqrt{10}$
1111	$-3/\sqrt{10}$	$-3/\sqrt{10}$

Tabla 3.7: Modulación 64-QAM. [1]

b(i)	I	Q	b(i)	I	Q
000000	$3/\sqrt{42}$	$3/\sqrt{42}$	100000	$-3/\sqrt{42}$	$3/\sqrt{42}$
000001	$3/\sqrt{42}$	$1/\sqrt{42}$	100001	$-3/\sqrt{42}$	$1/\sqrt{42}$
000010	$1/\sqrt{42}$	$3/\sqrt{42}$	100010	$-1/\sqrt{42}$	$3/\sqrt{42}$
000011	$1/\sqrt{42}$	$1/\sqrt{42}$	100011	$-1/\sqrt{42}$	$1/\sqrt{42}$
000100	$3/\sqrt{42}$	$5/\sqrt{42}$	100100	$-3/\sqrt{42}$	$5/\sqrt{42}$
000101	$3/\sqrt{42}$	$7/\sqrt{42}$	100101	$-3/\sqrt{42}$	$7/\sqrt{42}$
000110	$1/\sqrt{42}$	$5/\sqrt{42}$	100110	$-1/\sqrt{42}$	$5/\sqrt{42}$
000111	$1/\sqrt{42}$	$7/\sqrt{42}$	100111	$-1/\sqrt{42}$	$7/\sqrt{42}$
001000	$5/\sqrt{42}$	$3/\sqrt{42}$	101000	$-5/\sqrt{42}$	$3/\sqrt{42}$
001001	$5/\sqrt{42}$	$1/\sqrt{42}$	101001	$-5/\sqrt{42}$	$1/\sqrt{42}$
001010	$7/\sqrt{42}$	$3/\sqrt{42}$	101010	$-7/\sqrt{42}$	$3/\sqrt{42}$
001011	$7/\sqrt{42}$	$1/\sqrt{42}$	101011	$-7/\sqrt{42}$	$1/\sqrt{42}$
001100	$5/\sqrt{42}$	$5/\sqrt{42}$	101100	$-5/\sqrt{42}$	$5/\sqrt{42}$
001101	$5/\sqrt{42}$	$7/\sqrt{42}$	101101	$-5/\sqrt{42}$	$7/\sqrt{42}$
001110	$7/\sqrt{42}$	$5/\sqrt{42}$	101110	$-7/\sqrt{42}$	$5/\sqrt{42}$
001111	$7/\sqrt{42}$	$7/\sqrt{42}$	101111	$-7/\sqrt{42}$	$7/\sqrt{42}$
010000	$3/\sqrt{42}$	$-3/\sqrt{42}$	110000	$-3/\sqrt{42}$	$-3/\sqrt{42}$
010001	$3/\sqrt{42}$	$-1/\sqrt{42}$	110001	$-3/\sqrt{42}$	$-1/\sqrt{42}$
010010	$1/\sqrt{42}$	$-3/\sqrt{42}$	110010	$-1/\sqrt{42}$	$-3/\sqrt{42}$
010011	$1/\sqrt{42}$	$-1/\sqrt{42}$	110011	$-1/\sqrt{42}$	$-1/\sqrt{42}$
010100	$3/\sqrt{42}$	$-5/\sqrt{42}$	110100	$-3/\sqrt{42}$	$-5/\sqrt{42}$
010101	$3/\sqrt{42}$	$-7/\sqrt{42}$	110101	$-3/\sqrt{42}$	$-7/\sqrt{42}$
010110	$1/\sqrt{42}$	$-5/\sqrt{42}$	110110	$-1/\sqrt{42}$	$-5/\sqrt{42}$
010111	$1/\sqrt{42}$	$-7/\sqrt{42}$	110111	$-1/\sqrt{42}$	$-7/\sqrt{42}$
011000	$5/\sqrt{42}$	$-3/\sqrt{42}$	111000	$-5/\sqrt{42}$	$-3/\sqrt{42}$
011001	$5/\sqrt{42}$	$-1/\sqrt{42}$	111001	$-5/\sqrt{42}$	$-1/\sqrt{42}$
011010	$7/\sqrt{42}$	$-3/\sqrt{42}$	111010	$-7/\sqrt{42}$	$-3/\sqrt{42}$
011011	$7/\sqrt{42}$	$-1/\sqrt{42}$	111011	$-7/\sqrt{42}$	$-1/\sqrt{42}$
011100	$5/\sqrt{42}$	$-5/\sqrt{42}$	111100	$-5/\sqrt{42}$	$-5/\sqrt{42}$
011101	$5/\sqrt{42}$	$-7/\sqrt{42}$	111101	$-5/\sqrt{42}$	$-7/\sqrt{42}$
011110	$7/\sqrt{42}$	$-5/\sqrt{42}$	111110	$-7/\sqrt{42}$	$-5/\sqrt{42}$
011111	$7/\sqrt{42}$	$-7/\sqrt{42}$	111111	$-7/\sqrt{42}$	$-7/\sqrt{42}$

3.5. Procesos del Enlace Descendente

Para el enlace descendente se codifican los canales BCH, DL-SCH, PCH y MCH, en donde los procesos se reflejan en las Figuras 3.5 y 3.6

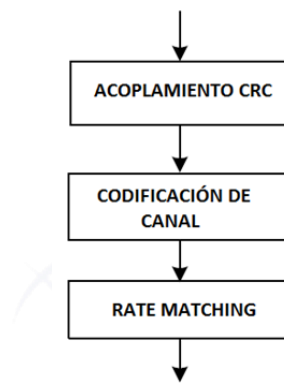


Figura 3.5: Pasos para la codificación del canal BCH. [1]

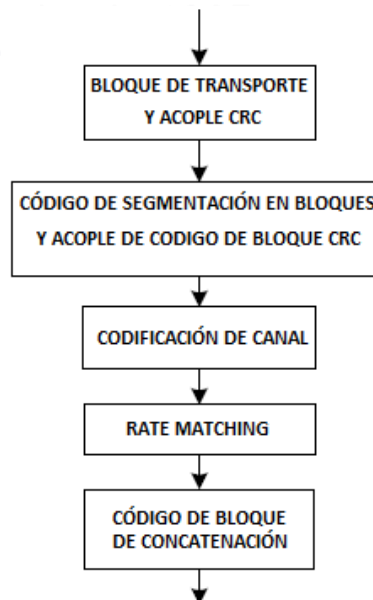


Figura 3.6: Procesamiento del bloque de transporte para DL-SCH, PCH y MCH. [1]

3.6. Procesos del Enlace Ascendente

Para el canal compartido de enlace ascendente, la estructura de procesamiento del canal de transporte (UL-SCH) para una celda de UL; se presenta en la Figura 3.7.

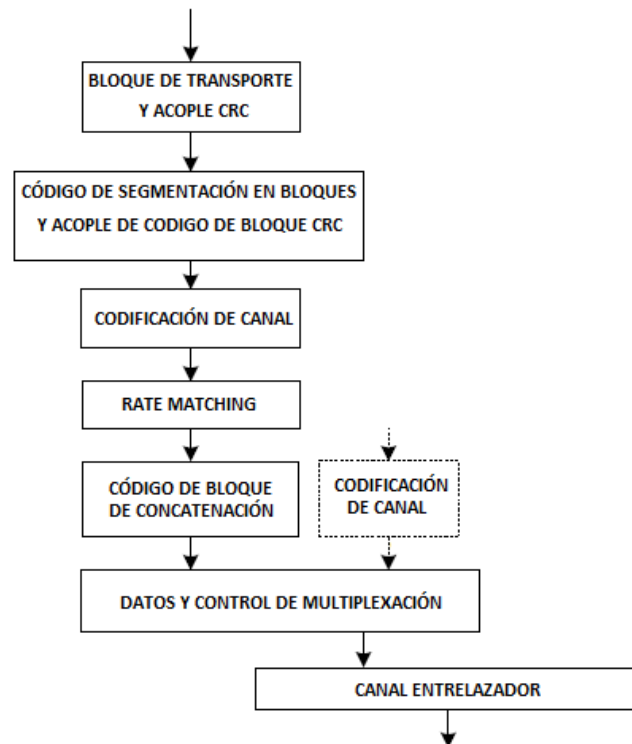


Figura 3.7: Procesamiento del Bloque de transporte para UL-SCH. [1]

3.7. Multiplexación de Frecuencias

En la red 4G LTE, los datos se manejan mediante tramas, la misma que puede ser de tres tipos, según la normativa 3GPP [1]; trama aplicable para FDD (*Frequency Division Duplex*), TDD (*Time Division Duplex*) y LAA (*License Assisted Access*).

Para el caso de estudio, la red LTE implementada en Ecuador, utiliza la trama correspondiente a FDD puesto que utilizan dos frecuencias, una para el enlace ascendente y otra para el enlace descendente. La trama para FDD se estructura de varios campos en el dominio de tiempo, en donde la unidad básica del

tiempo (T_s) viene dada por $T_s = 1/(15000 \times 2048)$ segundos; y las transmisiones son agrupadas en tramas de radio con una duración de $T_f = 307200 \times T_s = 10ms$.

Cada trama de radio se divide en 20 *slots*, en donde $T_{slot} = 15360 \times T_s = 0,5 ms$, y dos *slots* consecutivos se definen como una subtrama, con un tiempo de duración igual a $1ms = TTI$. (Figura 3.8)

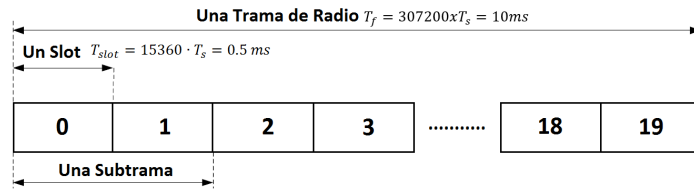


Figura 3.8: Estructura de la trama correspondiente a FDD. [1]

3.7.1. Estructura del Slot

Un *slot* presenta una estructura o una rejilla que se puede visualizar en las Figuras 3.9 y 3.10 tanto para el enlace ascendente como para el enlace descendente respectivamente.

Para el enlace ascendente se define como el número de símbolos SC-FDMA como N_{simb}^{UL} , un *resource block (RB)*, es igual a multiplicar el número de símbolos SC-FDMA, por el número de subportadoras, $RB = N_{simb}^{UL} \times N_{SubPor}^{RB}$. El número de símbolos SC-FDMA depende del prefijo cíclico al cual este configurado la red, y del número de subportadoras, los valores dependiendo del prefijo cíclico se muestran en la Tabla 3.8.

Tabla 3.8: Parámetros del bloque de recurso enlace ascendente. [1]

Configuración	Número de subportadoras	Número de símbolos SC-FDMA
Prefijo Cíclico Normal	12	7
Prefijo Cíclico Extendido	12	6

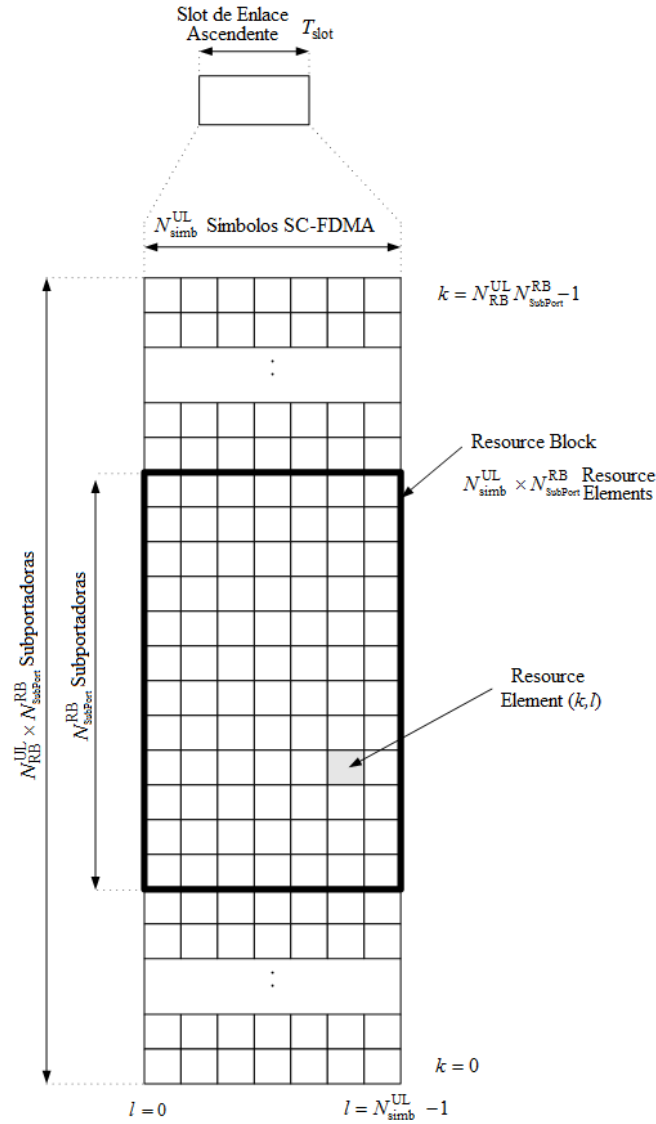


Figura 3.9: Estructura del slot para el enlace ascendente. [1]

Para el enlace descendente se define como el número de símbolos OFDMA como $N_{\text{symb}}^{\text{DL}}$, un *resource block*, es igual a multiplicar el número de símbolos OFDM, por el número de subportadoras, $RB = N_{\text{symb}}^{\text{DL}} \times N_{\text{SubPort}}^{\text{RB}}$. El número de símbolos OFDMA depende de la configuración del prefijo cíclico, Tabla 3.9.

El prefijo cíclico se emplea para evitar la interferencia intersímbolo (ISI) conservando así la ortogonalidad, el mismo es una copia de la última parte del símbolo OFDM, como se ve en la Figura 3.11.

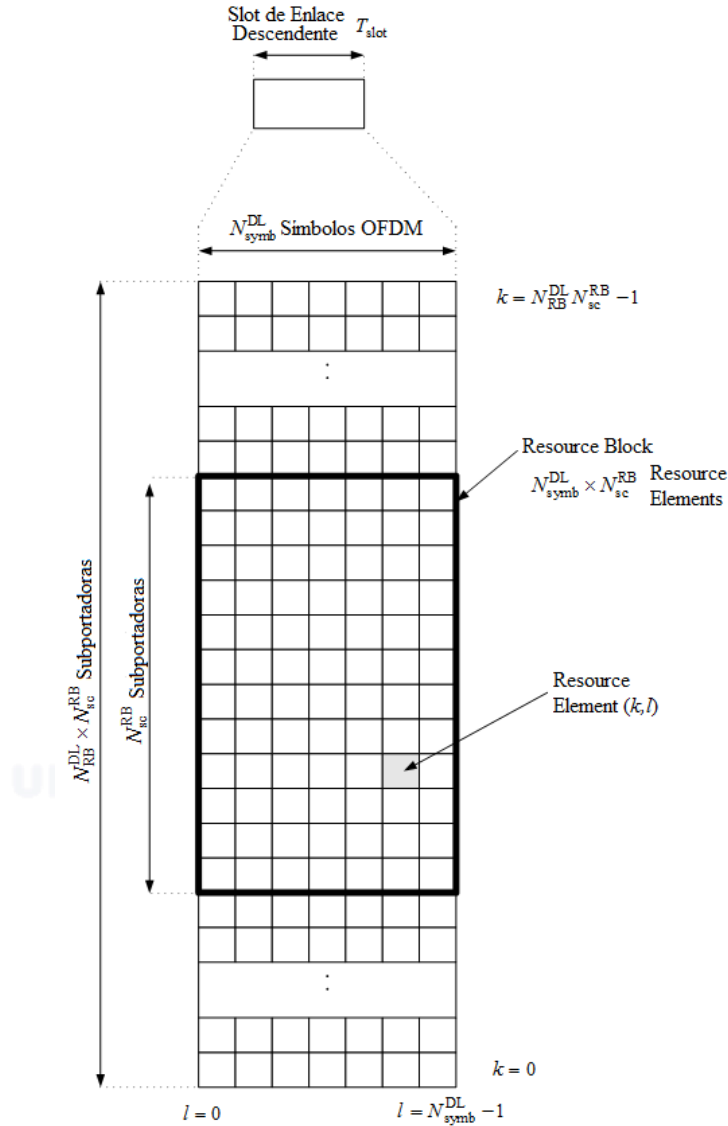


Figura 3.10: Estructura del *slot* para el enlace descendente. [1]

El número de *resource elements* (RE) por trama está dado por $N_{RE}^{Trama} = N_{RB} \times 12 \times N_{simb}^{TS} \times 20$, en donde N_{simb}^{TS} viene a ser el número de símbolos OFDM que dependen la configuración del prefijo cíclico.

El número de bits por RE está dado por $N_{bits}^{RE} = \log_2(M)$ bits y la tasa $V_{RE} = \frac{N_{bits}^{RE}}{0.5/N_{simb}^{TS}} = 2N_{simb}^{TS} \log_2 M$ Kb/seg.

Entonces se deduce que la tasa de RB viene dada por $V_{RB} = 24N_{simb}^{TS} \log_2 M$ Kb/seg y el número de bits por subtrama coincide con el número de bits, puesto que la trama dura 1ms, es decir $V_{SubTra} = N_{bits}^{SubTra}$.

Tabla 3.9: Parámetros del bloque de recurso enlace descendente. [1]

Configuración		Número de subportadoras	Número de símbolos OFDMA
Prefijo Cíclico Normal	$\Delta f = 15kHz$	12	7
Prefijo Cíclico Extendido	$\Delta f = 15kHz$		6
	$\Delta f = 7,5kHz$	24	3

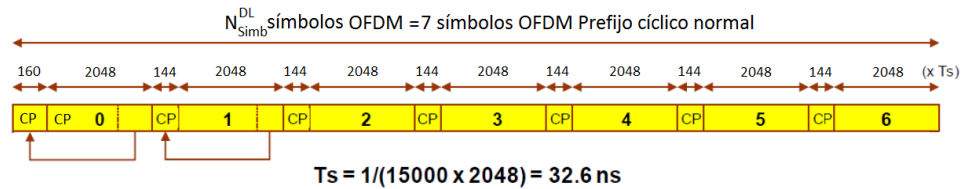


Figura 3.11: Prefijo cíclico evita ISI. [1]



UNIVERSIDAD DE CUENCA
desde 1867

Capítulo 4

Ejemplo Ilustrativo

4.1. Introducción

Este capítulo presenta un ejemplo como prueba de escritorio de cada uno de los procesos que abarca la capa física de la Red 4G LTE según lo revisado en el Capítulo 3.

Se va a transmitir la letra H codificada mediante el código universal ASCII, y se transforma a código binario, dando como resultado:

Tabla 4.1: Codificación letra H.

LETRA	CÓDIGO ASCII	CÓDIGO BINARIO
H	72	01001000

4.2. Canal de Codificación

Como se revisó en el Capítulo 2, la primera etapa es la codificación empezando por el codificador turbo, el mismo que se presenta a continuación. El código binario generado en la sección 4.1, ingresa en el codificador turbo que se muestra en la Figura 4.1.

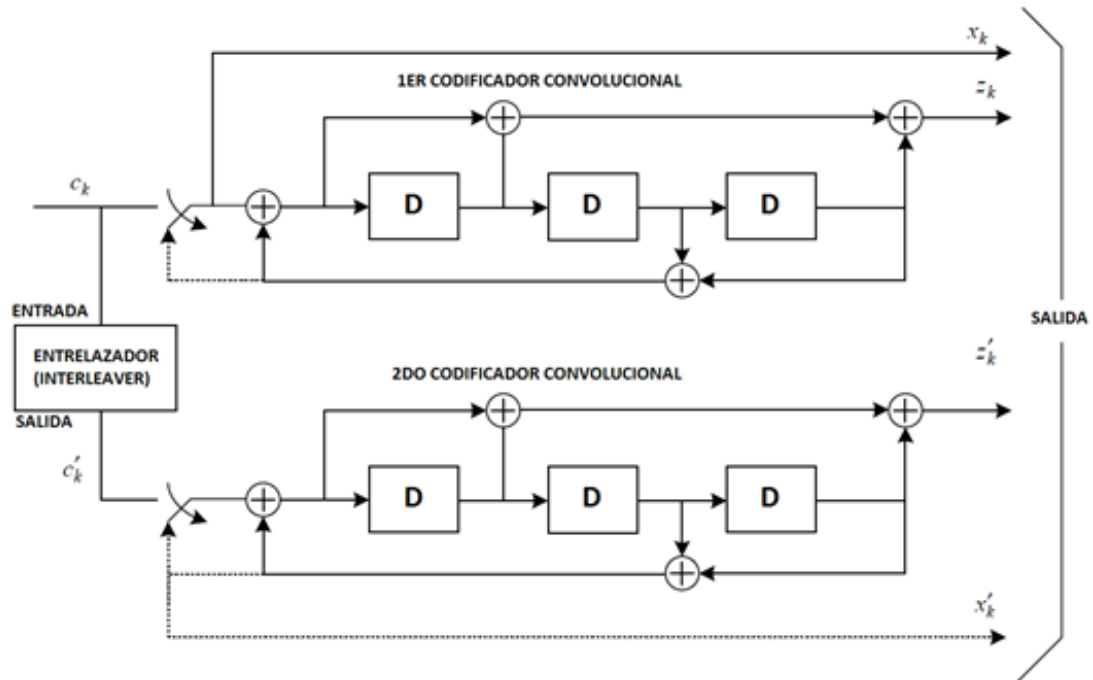


Figura 4.1: Codificador Turbo. [3]

4.2.1. Codificación CRC

Una vez realizada la codificación, a continuación sigue el proceso de CRC, en el cual a los 8 bits de la codificación ASCII se agregan 24 bits quedando el código 01001000000000000000000000000000, y para este ejemplo el polinomio generador es 1100001100100110011111011, entonces se ejecuta un proceso XOR, entre el polinomio generador y el código obtenido,

```

01001000000000000000000000000000
1100001100100110011111011
1000101100100110011111011
1100001100100110011111011
01001000000000000000000000000000
1100001100100110011111011
01010011001001100111110110
1100001100100110011111011
01100101011010101000011010
1100001100100110011111011
00001001111100110111000010000
1100001100100110011111011
0101110000010001011101011
  
```



Como resultado final se tiene la palabra codificada más el código obtenido en la última iteración XOR entre los dos códigos, el mismo es:

01001000101110000010001011101011

4.2.2. Segmentación de código de bloque

En este caso la longitud de la cadena de bits es 32, entonces según la Tabla 3.3, la cadena de bits tiene que ser de longitud mínima 40, por eso se completa la cadena con valores 0 adelante del resultado del CRC:

0000000001001000101110000010001011101011

4.2.3. Turbo Código

Se agregan 4 bits al final del código, puesto que estos servirán para poder obtener los 12 bits que se necesitan en Trellis.

00000000010010001011100000100010111010110000

Codificador Convolutacional I

Se obtiene un código, el mismo que es la entrada al primer codificador convolutacional de la Figura 4.1, el mismo que tiene la función de transferencia:

$$1 + D^2 + D^3 \quad (4.1)$$

En donde el primer valor por defecto de esta función es 0, entonces el proceso se visualiza en la Tabla, 4.2

La salida del primer codificador convolutacional esta conformada por:

**000000000000000000011000110000101110010101000
00010000110001011100101010001101101001000100**

Tabla 4.2: Codificador convolucional I.

ENTRADA	Proceso	SALIDA	ENTRADA	Proceso	SALIDA
0	0 0 0 0	0	0	0 1 1 1	0
0	0 0 0 1	1	1	1 1 1 0	0
0	0 0 1 1	0	1	1 1 0 1	0
0	0 1 1 0	1	1	1 0 1 0	0
1	1 1 0 1	0	0	0 1 0 0	0
1	1 0 1 0	0	1	1 0 0 0	1
0	0 1 0 1	1	0	0 0 0 1	1
1	1 0 1 1	1	0	0 0 1 0	1
0	0 1 1 1	0	0	0 1 0 0	0
1	1 1 1 0	0	1	1 0 0 1	0
1	1 1 0 1	0	0	0 0 1 0	1
1	1 0 1 0	0	0	0 1 0 0	0
0	0 1 0 0	0	1	1 0 0 0	1
1	1 0 0 0	1	0	0 0 0 0	0
0	0 0 0 1	1	0	0 0 0 0	0
0	0 0 1 0	1	0	0 0 0 0	0
0	0 1 0 0	0	0	0 0 0 0	0
1	1 0 0 0	1	0	0 0 0 0	0
0	0 0 0 0	0	0	0 0 0 0	0
0	0 0 0 0	0	0	0 0 0 0	0
0	0 0 0 1	1	0	0 0 0 0	0
0	0 0 1 1	0	0	0 0 0 0	0

Codificador Convolucional II

La entrada del segundo codificador convolucional de la Figura 4.1, el código pasa por un *interleaver*, el mismo que según la 3GPP está dado por la función:

$$\Pi(i) = (f_1 i + f_2 i^2) \text{ MOD } (K) \quad (4.2)$$

En donde el operador MOD, es igual al residuo de la división de $(f_1 i + f_2 i^2)$ para K; el mismo que es la posición del bit del antiguo código, con el cual se genera la entrada al segundo codificador convolucional.

Según la tabla del 3GPP, se tiene que para un valor de $K = 40$; $f_1 = 3$, y $f_2 = 10$; reemplazando estos datos en la Ecuación 4.2 nos da como resultado la Ecuación 4.3. El resultado total del proceso *interleaver* se refleja en la Tabla 4.3.

$$\Pi(i) = (3 i + 10 i^2) \text{ MOD } (40) \quad (4.3)$$

Tabla 4.3: Entrelazador (*Interleaver*).

i	Pi(i)	BINARIO	i	Pi(i)	BINARIO
0	0	0	20	20	1
1	13	0	21	33	1
2	6	0	22	26	1
3	19	1	23	39	1
4	12	1	24	32	1
5	25	0	25	5	0
6	18	1	26	38	1
7	31	0	27	11	0
8	24	0	28	4	0
9	37	0	29	17	0
10	30	1	30	10	0
11	3	0	31	23	0
12	36	1	32	16	1
13	9	1	33	29	0
14	2	0	34	22	0
15	15	0	35	35	0
16	8	0	36	28	0
17	21	0	37	1	0
18	14	0	38	34	1
19	27	0	39	7	0



CAPÍTULO 4. EJEMPLO ILUSTRATIVO

Entonces la entrada al codificador convolucional II, está dada por:

00011010001011000000111110100000100000100000

Los 4 últimos bits se agregan para utilizar posteriormente en Trellis. La función de transferencia de este codificador, se define mediante la Ecuación 4.4; y el proceso se presenta en la Tabla 4.4.

$$1 + D + D^3 \quad (4.4)$$

Tabla 4.4: Codificador convolucional II.

ENTRADA	Proceso	SALIDA	ENTRADA	Proceso	SALIDA
0	0 0 0 0	0	1	1 1 0 0	0
0	0 0 0 0	0	1	1 0 0 0	1
0	0 0 0 1	1	0	0 0 0 0	0
0	0 0 1 0	0	0	0 0 0 0	0
0	0 1 0 0	1	0	0 0 0 0	0
1	1 0 0 0	1	0	0 0 0 1	1
0	0 0 0 0	0	0	0 0 1 1	1
0	0 0 0 0	0	0	0 1 1 0	1
0	0 0 0 1	1	1	1 1 0 1	1
0	0 0 1 0	0	1	1 0 1 0	1
0	0 1 0 0	1	0	0 1 0 0	1
1	1 0 0 0	1	1	1 0 0 0	1
0	0 0 0 0	0	0	0 0 0 1	1
0	0 0 0 0	0	0	0 0 1 0	0
0	0 0 0 1	1	0	0 1 0 1	0
0	0 0 1 0	0	1	1 0 1 1	0
0	0 1 0 1	0	0	0 1 1 0	1
1	1 0 1 1	0	1	1 1 0 0	0
0	0 1 1 1	0	1	1 0 0 0	1
1	1 1 1 1	1	0	0 0 0 0	0
1	1 1 1 1	1	0	0 0 0 0	0
1	1 1 1 0	0	0	0 0 0 0	0

La salida del segundo codificador convolucional es:

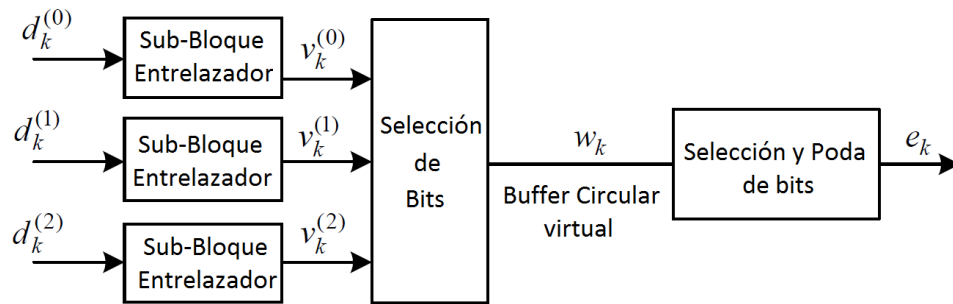
00000011011001000010111011111010100000001101
01111100010000100000111000100000111000100000

Salida Turbo Código

Entonces la salida del turbo código, tomando en cuenta los 12 últimos bits con la terminación de Trellis se presenta a continuación.

```
000000000000100000100000000011100101110100101
10111110001001001010000000110010001100000100
11110000101101100001110010101101010001000100
```

4.3. Adaptación de Velocidad



$d_1^{(0)} = 000000000010010001011100000100010111010110001$

$d_1^{(1)} = 000000000010100111000000100101110000011001000$

$d_1^{(2)} = 00010100011111111000100110000100110100110100$

Se determina el número total de bits del mensaje D de acuerdo a la Ecuación 3.10. La salida del sub-bloque entrelazador, el número de columnas C es igual a 32 y el número de filas F , esta dado por $F = D/C$, en donde $D = 44$ y F igual a 2 aproximadamente.

$$D = \text{Bits Mensaje} + 24 \text{ Bits CRC} + 4 \text{ Bits Trellis} + \text{Nand Asignadas}$$

$$D = 8 + 24 + 4 + 8$$

Se tiene que $K_{\Pi} = C \times F = 64$ entonces se agregan 20 bits ficticios al principio de la matriz que se genera.

Matriz $d_1^{(0)}$

```
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN000000
00010010001011100000100010111010110001
```

Matriz $d_1^{(1)}$

NNNNNNNNNNNNNNNNNNNNNNNNNNNN000000
00010100111000000100101110000011001000

Matriz $d_1^{(2)}$

NNNNNNNNNNNNNNNNNNNNNNNNNNNN000101
00011111111000100110000100110100110100

Mediante el polinomio 3.13, se obtiene la permutación para cada matriz generada en el punto anterior.

Permutación $d_1^{(0)}$

NNN0N0N0NNN0N0N0NNN0N0N1NNN0
N0N010111100010111100000010000011001

Permutación $d_1^{(1)}$

NNN0N0N0NNN0N0N0NNN0N0N1NNN0
N0N101011001110000100101000010100000

Una vez realizada la permutación para $d_1^{(0)}$ y $d_1^{(1)}$, se obtienen las salidas $v_1^{(0)}$ y $v_1^{(1)}$.

Salida $v_1^{(0)}$

N1N0N101N101N000N0N1N001N101N100
N0N0N000N001N010N0N0N001N100N001

Salida $v_1^{(1)}$

N0N1N001N100N001N1N1N000N000N100
N0N1N001N000N010N1N0N100N000N010

Para obtener la salida $v_1^{(2)}$, se aplica la Ecuación 3.15, mediante la cual se determina las posiciones de la misma.

$$\Pi(k) = \left(P \left(\left\lfloor \frac{k}{F} \right\rfloor \right) + C \times (k \bmod F) + 1 \right) \bmod K_{\Pi}$$

En donde para $\Pi(0)$ se tiene que es igual al 1, el mismo que corresponde a la posición 1 de la matriz $d_1^{(2)}$, que es igual a N . Para $\Pi(1)$ se tiene que es igual a 33, el mismo que corresponde a la posición 33 de la matriz $d_1^{(2)}$, que es igual a 1.

$$\Pi(0) = \left(\left\lfloor P \left(\frac{0}{2} \right) \right\rfloor + 32 \times 0 + 1 \right) \bmod (64)$$

$$\Pi(0) = 1$$

$$\Pi(1) = \left(\left\lfloor P \left(\frac{1}{2} \right) \right\rfloor + 32 \times 1 + 1 \right) \bmod (64)$$

$$\Pi(1) = 33$$

Se realiza este proceso hasta $\Pi(k - 1)$, y se obtienen las posiciones para la salida $v_1^{(2)}$.

Posiciones $\Pi(k)$

1 33 17 49 9 41 25 57 5 37 21 53 13 45 29 61 3 35 19 51 11 43
27 59 7 39 23 55 15 47 31 63 2 34 18 50 10 42 26 58 6 38 22
54 14 46 30 62 4 36 20 52 12 44 28 60 8 40 24 56 16 48 32 0

Salida $v_1^{(2)}$

N1N1N010N001N011N1N0N101N011N010
N1N0N001N000N010N101N100N100N01N

Buffer Circular

Se calcula $K_w = 3K_\Pi = 3(64) = 192$ y mediante las ecuaciones definidas en la Sección 3.2.2, se conoce la salida w_k

$$\begin{pmatrix} w_k = v_k^{(0)} \\ w_{K_\Pi+k} = v_k^{(1)} \\ w_{2K_\Pi+k} = v_k^{(2)} \end{pmatrix} k = 0, \dots, K_\Pi - 1$$

Salida w_k

N1N0N101N101N000N0N1N001N101N100
N0N0N000N001N010N0N0N001N100N001
NN01NN11NN000110NN100001NN000111
NN11NN10NN010001NN000101NN100100
NN01NN10NN000011NN000000NN001100
NN11N001NN110000NN010000NN00110N

Selección de bits e_k

Como se vio en la Sección 3.2.2, se realizan algunas consideraciones para obtener la salida e_k del *rate matching*.

- $N_{soft} = 3654144$, puesto que se elige un UE de categoría 6 o 7, según la 3GPP. entonces $K_C = 2$.
- $C = 1$ número de códigos de bloque.
- $K_{MIMO} = 1$
- $M_{DLHARQ} = 8$ es número máximo de procesos HARQ.
- $M_{limit} = 8$
- Entonces aplicando la Ecuación 3.18, se obtiene $N_{IR} = 228384$.
- Y aplicando la Ecuación 3.16, el valor de $N_{cb} = 192$, para el enlace descendente
- $N_L = 2$.
- $G = 132$,
- Para este ejemplo, se realiza la modulación QPSK, en donde $Q_m = 2$.
- Entonces $G' = 33$
- Y $\gamma = 0$. Por la ecuación $\gamma = G' \bmod C$
- Se deduce el valor de $E = 132$, y para el índice de redundancia $rv_{idx} = 0$
- Se obtiene el valor de $k_0 = 4$ mediante la Ecuación 3.20

Y se obtiene la salida e_k a partir de w_k .

Salida e_k

```
101101000010011011000000000101000
001100001011100011010000100011111
100100010001011001000110000011000
000001100110011100000100000011010
```



Salida 1

```
000000100001101000010010011110100
010010110010101000000110101011000
010010011110100011001111001101000
000110101011101011011010010100011
```

Salida 2

```
011000100011101001011000010111000
111111011101010100101101110110100
011111100000101010111111010111010
001010011000001100011001111011110
```

Secuencia Gold

```
101100000001000000100101000100110
010110110111111110010101101110110
000110111111000100111000001101001
000110011001110011100001110111110
```

Una vez obtenida la secuencia *gold*, se realiza la operación XOR con la salida del *rate matching*, para obtener finalmente la salida del *scrambling*.

Salida Scrambling

```
110101000000011010001010001100100
100001100100011111111011111110011
101001101111010000110110011001010
001101010101111011000111101100111
```

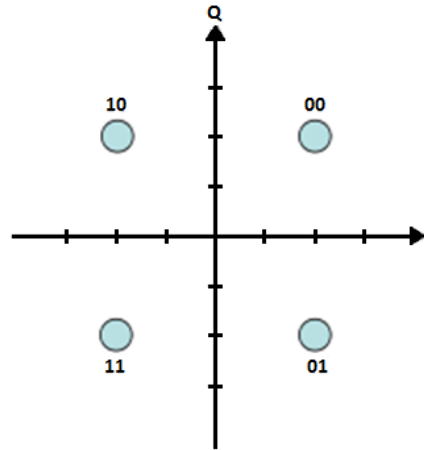
4.5. Modulaciones

Para la modulación QPSK, se tiene 2^n posibles valores de mapeo, siendo $n = 2$ el número de bits por símbolo a transmitir.

Segun la Sección 3.4 el mapeo está denifido por:

b(i)	I	Q
00	$1/\sqrt{2}$	$1/\sqrt{2}$
01	$1/\sqrt{2}$	$-1/\sqrt{2}$
10	$-1/\sqrt{2}$	$1/\sqrt{2}$
11	$-1/\sqrt{2}$	$-1/\sqrt{2}$

Y la constelación se presenta en el siguiente diagrama.



4.6. Multiplexación de Frecuencias (OFDM)

Dentro de OFDM se realizan los cálculos de las capacidades de los elementos de las señales de LTE [6].

- **Número de elemento de recurso por trama:** Cabe mencionar que cada trama posee 20 *slots* y 12 subportadoras, entonces el número de RE por trama esta dado por:

$$N_{trama}^{RE} = N_{RB} \cdot 12 \cdot N_{simbolo}^{slot} \cdot 20$$

En donde $N_{simbolo}^{slot}$, depende del prefijo cíclico, en este caso, se utiliza prefijo cíclico normal, que corresponde a 7 símbolos por *slot* entonces:

$$N_{trama}^{RE} = 1 \cdot 12 \cdot 7 \cdot 20 = 1680$$

- **Número de bits y velocidad por elemento de recurso:** Para poder realizar este cálculo, se necesita saber el número de bits por símbolo de la modulación.

$$N_{bits}^{RE} = \log_2 M \text{ (bits)}$$

En este ejemplo la modulación utilizada es QPSK, entonces:

$$N_{bits}^{RE} = \log_2(4) = 2 \text{ bit}$$

La tasa binaria o velocidad se calcula mediante:

$$R_{RE} = 2 \cdot N_{simbolo} \cdot \log_2 M \text{ (Kb/seg)}$$

$$R_{RE} = 2 \cdot 7 \cdot \log_2 4 = 28 \text{ (Kb/seg)}$$

- **Velocidad por bloque de recurso:** La velocidad o tasa binaria por bloque de recurso está dada por:

$$R_{RB} = 12 \cdot R_{RE} = 12 \cdot 28 = 336 \text{ Kb/seg}$$

- **Bits y velocidad por subtrama:** El número de bits y la velocidad por subtrama coinciden, puesto que la duración de la misma es de 1 ms, entonces:

$$N_{bits}^{Subtrama} = N_{RB} \cdot 12 \cdot 2 \cdot N_{TS}^{simbolo} \cdot \log_2 M$$

$$N_{bits}^{Subtrama} = 1 \cdot 12 \cdot 2 \cdot 7 \cdot 2 = 336 \text{ bits}$$

Entonces la tasa de bits:

$$R_{Subtrama} = N_{bits}^{Subtrama} = 336 \text{ (Kb/seg)}$$

La Figura 4.2 muestra la rejilla OFDM con cada uno de sus canales y la asignación de recursos.

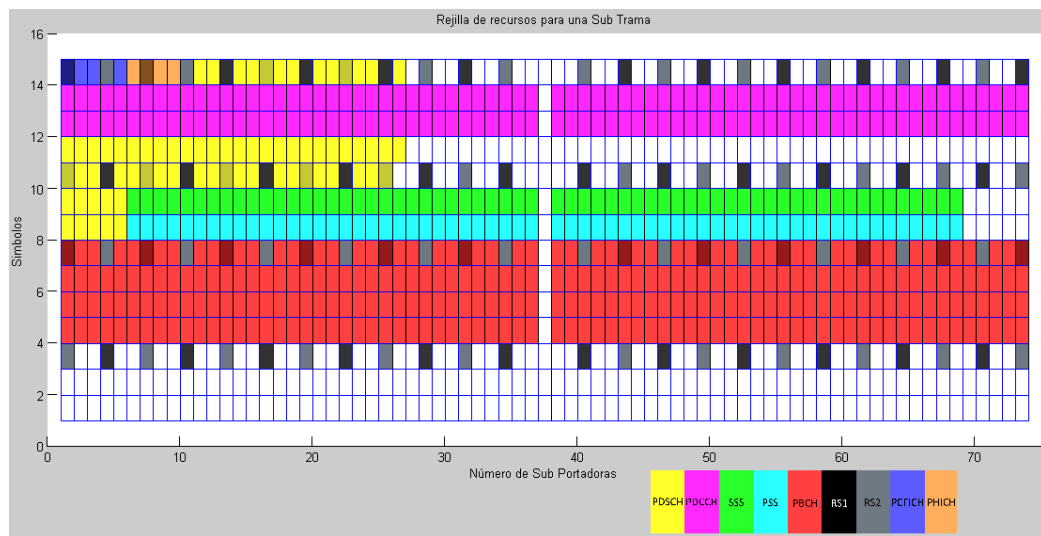


Figura 4.2: Rejilla de Recursos para una Subtrama.

4.7. MIMO

La información se mapea en dos capas, como se puede observar en la Figura 4.3. En donde se tiene 2 antenas en el emisor y dos antenas en el receptor, una configuración MIMO 2x2.

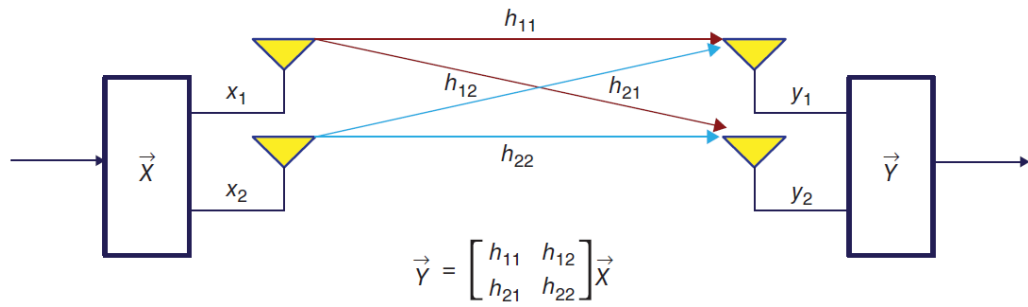


Figura 4.3: Técnicas Multiantenas. [16]

Como se puede observar en la Figura 4.3, los símbolos x_1 y x_2 , se transmiten por el canal radio móvil, en donde la matriz H , corresponde a la respuesta del canal que une la antena transmisora con la antena receptora, siendo los caminos de propagación creados por las 4 antenas presentes [12], entonces la señal que recibe la antena 1, y_1 , y la señal que recibe la antena 2, y_2 , están dadas por:

$$y_1 = h_{1,1}x_1 + h_{1,2}x_2$$

$$y_2 = h_{2,1}x_1 + h_{2,2}x_2$$

La señal enviada se representa en forma matricial [11] de la siguiente manera:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Entonces los símbolos quedan mapeados según se muestra en la Figura 4.4.

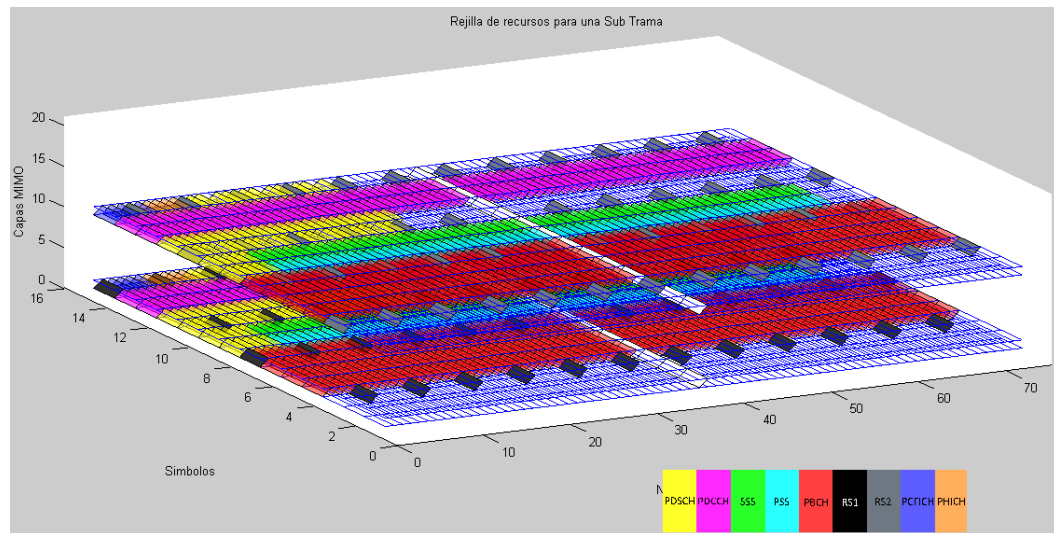


Figura 4.4: Mapeo de los símbolos.

4.8. Receptor

Para el proceso de demodulación, se le agrega un ruido al mensaje para simular la pérdida de paquetes. Con un $E_b/N_0=15$, y para una potencia=20, los bits demodulados recibidos con ruido son:

```
110101000000011010001010001100100
100001100100011111111011111110011
101001101111010000110110011001010
001101010101111011000111101100111
```

El proceso de *Des-Scrambling*, se ejecuta de la misma manera que el *scrambling* pues es un proceso ortogonal, logrando el resultado siguiente:

```
101101000010011011000000000101000
001100001011100011010000100011111
100100010001011001000110000011000
000001100110011100000100000011010
```


El proceso inverso al *rate matching* (*Rate Dematch*), presenta como resultado:

**000000000001000001000000000111001
011101001011011111000100100101000
000011001000110000010011110000101
101100001110010101101010001000100**

Para obtener la salida del turbo decodificador se realiza el proceso de Trellis para los 132 bits. Al tomar los 3 primeros bits, del *rate dematch*, 000, se compara con las salidas a las cuales puede pasar en caso de que el bit procesado haya sido un 0, 000 y 001, y con las salidas en caso de que el bit sea 1, 111 y 110, en donde al comparar las métricas con cada uno de las salidas mencionadas se tiene que para la salida 000 la métrica es 0, para la salida 001, la métrica es 1, para la salida 111 la métrica es 3, y para la salida 001 la métrica es 1. Entonces se elige la métrica de menor valor claramente, y el estado queda en 000, y el bit que se obtiene de salida es un 0. De manera igual se procede para los 6 siguientes bits, en donde la métrica para el estado 000, resulta en 0, generando como salida dos ceros respectivamente. Para los bits 001, se compara a que salidas puede cambiar, en caso de que haya sido un 0, las salidas pueden ser 001 y 000; mientras que para el bit 1, las salidas son 110, 111, en este caso las métricas para la salida 001, es 0, para la salida 000 es 1, para la salida 110 la métrica es 3 y para la salida 111 la métrica es 2. Entonces se deduce que según la métrica menor la entrada del bit fue un 0. Se sigue de esta manera, obteniendo los primeros 8 bits de salida del decodificador, que vienen a ser 0. Para la entrada 111, se compara con las salidas dependiendo del bit de entrada, estos son, si el bit de entrada fue un cero, las salidas pueden ser 000 y 001, mientras que si la entrada fue el bit 1, las salidas son 111 y 110, al obtener la métrica se tiene que para la salida 000 la métrica es 3, para la salida 001, la métrica es 1, para la salida 111 la métrica es 0 y para la salida 110, la métrica es 1. Entonces la métrica menor indica que el bit es 1.

Salida parcial obtenida mediante la decodificación **000000001**, Figura 4.5. El proceso se realiza de la misma manera para los demás bits, dando como resultado del turbo decodificador la cadena de bits:

0000000001001000101110000010001011101011

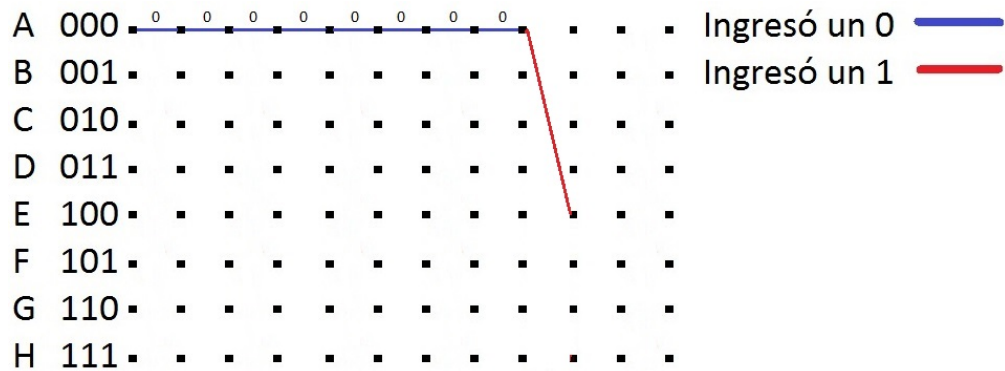


Figura 4.5: Diagrama de Trellis para la salida parcial.

En el proceso de CRC, se realiza de la misma manera que en el emisor, pero el residuo que devuelve es igual a cero, lo cual garantiza que el mensaje llegó sin error, de esta forma se envía un ACK. Posterior se eliminan los 8 bits de relleno al inicio y los 24 bits de paridad al final.

01001000

Al transformar de binario a código ASCII, se observa que la letra obtenida es una H.

Capítulo 5

Descripción del Software

5.1. Introducción

El software se desarrolló en la plataforma MATLAB, el mismo que utiliza un lenguaje propio, lenguaje M. Esta plataforma es un software muy usado en universidades y centros de investigación, el cual a sido muy utilizado a lo largo de nuestra carrera. MATLAB es un software matemático ideal para trabajar con vectores y matrices, siendo idóneo para la implementación de los algoritmos que se desarrollaron para cada uno de los procesos de la capa física de LTE. Adicional, cuenta con un módulo para la creación de interfaces de usuario GUI, la misma que resulta muy útil para la visualización de cada uno de los procesos y resultados obtenidos; y de esta manera realizar un software didáctico como se planteó.

MATLAB dispone de herramientas adicionales como Simulink (Plataforma de Simulación Multidominio) y paquetes “Toolbox”, como “*Communications System Toolbox*”, con estas herramientas se pudo comparar procesos y resultados, y así validar los algoritmos programados en este proyecto de titulación.

5.2. Estructura del software

El software cuenta con un menú principal, en el mismo se observa tres tipos de simulación.

1. Simulación paso a paso del enlace descendente.
2. Simulación paso a paso del enlace ascendente.
3. Simulación automática.

En la Figura 5.1, se presenta el diagrama de bloques general del software desarrollado.

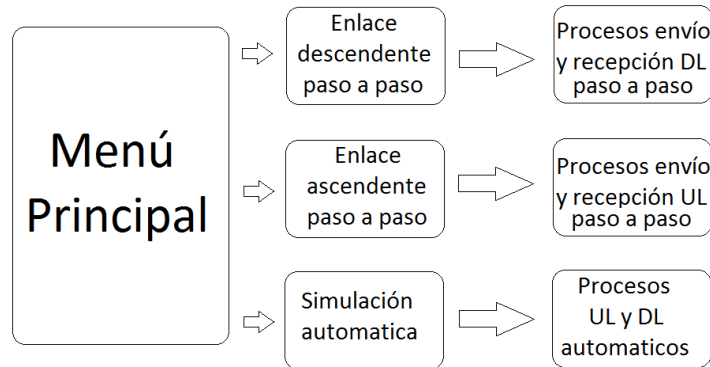


Figura 5.1: Diagrama de bloques del software.

Los procesos para el enlace ascendente como el enlace descendente se presentan desarrollados paso a paso, la estructura de los mismos se visualizan en las Figuras 5.2, 5.3 y 5.4.

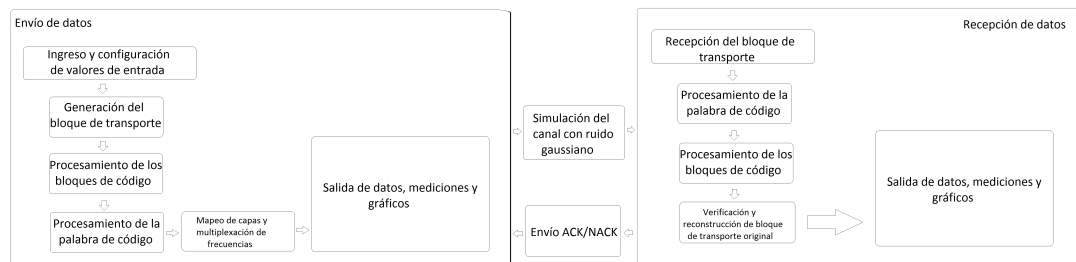


Figura 5.2: Procesos envío y recepción de datos.

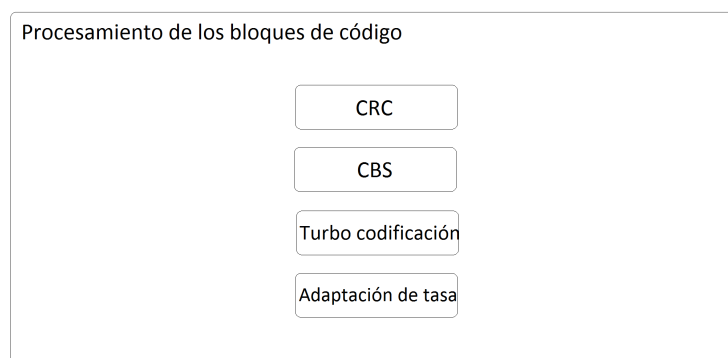


Figura 5.3: Procesos bloques de código.

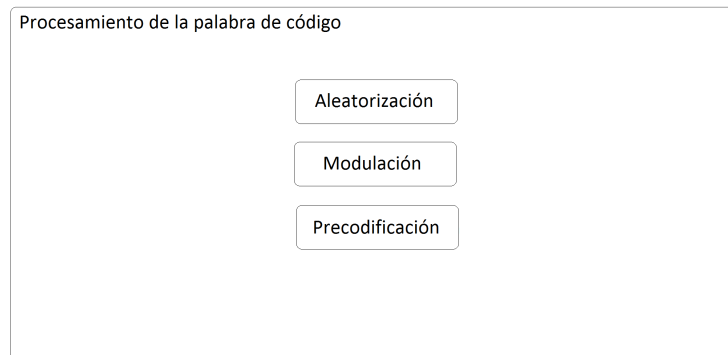


Figura 5.4: Procesos palabra de código.

El software permite realizar configuraciones de acuerdo a la simulación que se desea, dependiendo de los parámetros ingresados por el usuario. El usuario puede configurar los parámetros siguientes:

- Tipo de información (Bloque de transporte).
- EbNo (dBm).
- Potencia(W).
- Número de usuarios (1, SU-MIMO o 2, MU-MIMO).
- Adaptación de tasa (Rate).
- Canal (PDSCH o PBCH, solo en DL).
- Banda de frecuencia.
- Precodificación con redundancia o aumento de tasa.
- Modulación.
- Valor de desfase o desincronización (Mediante canal SSS y PSS).
- PLI y PLCIG (identidad de la celda).

Las salidas que el software procesa y permite visualizar dentro de cada simulación son:

- Bits de información en cada proceso.
- Detalle de cada proceso ejecutado paso a paso.
- Número de bits generados para el bloque de transporte.
- Número de bits en cada proceso.
- Número de bits por símbolo.
- Banda de frecuencia.
- Tiempo de bit.
- Bloque de código.

- Bloque de transporte.
- Capacidad bruta.
- Tasa canal PDSCH.
- Número de bits recibidos.
- CQI (UE).
- Envío de ACK o NACK.
- Residuo CRC.
- BER en cada proceso en la recepción de los datos.
- Gráficos:
 - Gráfico del tren de bits de cada proceso.
 - Gráfico de la celda en la que se ubica el usuario.
 - Gráfico de capas MIMO utilizadas.
 - Gráfico de la rejilla OFDM o SC-FDM, con cada uno de los canales utilizados en DL o UL respectivamente, lo RE, RB, *slots*, y subportadoras utilizadas.
 - Gráficos de las subportadoras utilizadas y moduladas.
 - Gráfico de toda una trama y subtramas utilizadas.
- Diagramas:
 - Diagrama de constelación de la modulación seleccionada.
 - Esquema de modulación a detalle comparando diagramas de constelación, tren de bits y subportadora.
 - Diagrama de constelación con el ruido del canal.

El software tiene valores predefinidos, los mismos no pueden ser configurados, estos valores son: el prefijo cíclico, el cual está configurado como normal (7 símbolos por *slot*); los valores para canales de sincronización o de formato, entre otros. Adicionalmente el software permite guardar los resultados obtenidos en cada simulación, junto con los valores configurados en un archivo de Excel.

5.3. Desarrollo e implementación de algoritmos

El software cuenta con más de cincuenta *scripts* o archivos de MATLAB (.m), cada uno de ellos cuenta con varios algoritmos, los cuales en su totalidad, fueron desarrollados por los autores de esta tesis.

Los *scripts* principales son aquellos en los que se implementaron los procesos propios de la capa física de LTE, tanto para el envío como para la recepción de la información, en los enlaces ascendente y descendente. Cada uno estos procesos están implementados mediante algoritmos desarrollados a partir de un estudio profundo de las diferentes especificaciones 3GPP, las mismas que explican de forma detallada los procesos que abarca esta tecnología. Los procesos especificados en la norma 3GPP son los siguientes:

1. CRC.
2. Segmentación de bloque de código.
3. Turbo codificación.
4. Adaptación de tasa.
5. Aleatorización.
6. Modulación y demodulación .
7. Técnicas MIMO y Precodificación.
8. OFDM.
9. SC-FDM.

En la recepción los procesos como el CRC, segmentación de bloque de código, adaptación de tasa, aleatorización, son exactamente los mismos o inversos a los realizados en el envío de datos. Dentro de la recepción, está desarrollado la decodificación de Trellis, así como la recuperación de la información de las antenas MIMO y demodulación OFDM o SC-FDM; en donde se puede visualizar a detalle cada uno de ellos mediante la interfaz gráfica.

5.4. Validación del software

Para realizar la validación del software desarrollado, se compara cada uno de los procesos desarrollados, con el módulo *comm* propio de MATLAB. A continuación se presenta la validación desarrollada.

5.4.1. Codificación CRC

En la Figura 5.5, se presenta como entrada la letra *H*, codificada en código ASCII [01001000], y la salida obtenida luego del proceso CRC. Adicional se puede observar la gráfica del tren de *bits* así como la longitud del mismo.

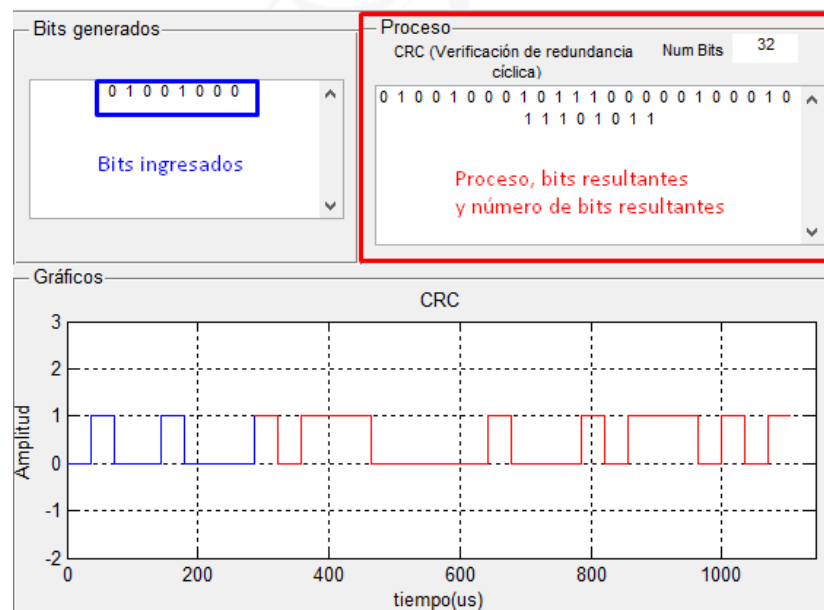


Figura 5.5: Codificación CRC, software implementado.

En la Figura 5.6, se puede observar el proceso ejecutado con el módulo de comunicaciones de MATLAB, *comm.CRCGenerator*, en donde se configura el vector del polinomio generador especificado en la 3GPP, para bloques de transporte menores a 6144 *bits*.


```
>> x=[0 1 0 0 1 0 0 0];
>> hGen = comm.CRCGenerator('Polynomial',[1 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 1 1 1 0 1 1]);
>> codeword = step(hGen, x);
>> codeword'
```

ans =
Columns 1 through 24
0 1 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 1 0 0 0 1 0
Columns 25 through 32
1 1 1 0 1 0 1 1

Figura 5.6: Codificación CRC, MATLAB.

Al comprar los resultados obtenidos en las Figuras 5.5 y 5.6, se puede observar que los resultados son los mismos, validando de esta manera el proceso desarrollado.

5.4.2. Segmentación de bloque de código

En este proceso se divide el bloque de transporte, en bloques de código con un tamaño máximo 6144 *bits* y mínimo de 40 *bits*, para este proceso MATLAB no posee un comando propio, sin embargo en el caso de este ejemplo el bloque de código tiene un tamaño de 32 *bits* y se completa hasta 40 *bits*, en la Figura 5.7 se muestra el proceso realizado en el software implementado.



Figura 5.7: Segmentación de bloque de código.

5.4.3. Turbo código

Este proceso se realiza mediante un turbo codificador con tasa 1/3, este es el resultado de la unión de dos codificadores convolucionales; la entrada para el primer codificador convolucional es el tren de *bits* recibido de la segmentación de código, y para el segundo codificador convolucional, el tren de *bits* de la segmentación de código pasa por un entrelazador, en la Figura 5.8 se observan los *bits* de entrada para cada codificador convolucional, así como las salidas de cada uno de ellos y la salida total del turbo codificador, que es la combinación de las dos salidas anteriores.

La entrada del primer codificador convolucional es 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 1 0 0 0 1 0 1 1 1 0 1 0 1 1 0 0 0 0, en donde los cuatro últimos *bits* se agregan para las salidas de Trellis. Se obtiene la salida de este codificador convolucional: 0 1 1 0 0 0 1 1 0 0 0 0 1 0 1 1 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 1 1 1 0 0 1 0 1 0 1 0 0 0 1 1 0 1 1 0 1 0 0 1 0 0 0 1 0 0.

Para el segundo codificador convolucional se calculan las posiciones de entrelazado, las mismas son: 1 14 7 20 13 26 19 32 25 38 31 4 37 10 3 16 9 22 15 28 21 34 27 40 33 6 39 12 5 18 11 24 17 30 23 36 29 2 35 8, en donde el tren de *bits* que ingresa al codificador es: 0 0 0 1 1 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0, y se obtiene la salida del codificador: 0 0 0 0 0 0 1 1 0 1 1 0 0 1 0 0 0 0 1 0 1 1 1 0 1 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 1 1 0 1 0 1 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0.

Al combinar las salidas obtenidas, se determina la salida del turbo codificador: 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 0 1 0 0 1 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 1 0 1 1 0 1 1 0 1 1 0 0 0 0 1 1 1 0 0 1 0 1 0 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0.



Figura 5.8: Turbo código, software implementado.

```
>> u=[0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 1 0 0 0 1 0 1 1 1 0 1 0 1 1 0 0 0 0]; Entrada 1er codificador
>> hConEnc = comm.ConvolutionalEncoder('trellisStructure',poly2trellis(4,[10,13]));
>> y=step(hConEnc,u);
>> y'
```

ans =

Columns 1 through 24
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1

Columns 25 through 48
1 0 0 0 0 1 0 1 1 1 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 1

Columns 49 through 72
0 0 0 0 1 1 0 0 0 1 0 1 1 1 0 0 1 0 1 0 1 0 0 0 0

Columns 73 through 88
1 1 0 1 1 0 1 0 0 1 0 0 0 1 0 0

Salida 1er codificador

Figura 5.9: Primer codificador convolucional, MATLAB.

```
>> u=[0 0 0 1 1 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0]; Entrada 2do codificador
>> hConEnc = comm.ConvolutionalEncoder('TrellisStructure',poly2trellis(4,[15,10]));
>> y=step(hConEnc,u);
>> y'
```

```
ans =
Columns 1 through 24|
    0    0    0    0    0    0    1    1    0    1    1    0    0    1    0    0    0    0    1    0    1    1    1    0
Columns 25 through 48
    1    1    1    1    1    0    1    0    1    0    0    0    0    0    0    0    1    1    0    1    0    1    1    1
Columns 49 through 72
    1    1    0    0    0    1    0    0    0    0    1    0    0    0    0    0    1    1    1    0    0    0    1    0
Columns 73 through 88
    0    0    0    0    1    1    1    0    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0
Salida 2do codificador
```

Figura 5.10: Segundo codificador convolucional, MATLAB.

Finalmente la salida total del turbo codificador se obtiene de combinar estas dos salidas, mediante *bit* sistemático, primer *bit* de paridad, segundo *bit* de paridad, en donde el *bit* sistemático son los *bits* de las posiciones impares de la salida del primer codificador convolucional, el primer *bit* de paridad son todos los *bits* de las posiciones pares de la salida del primer codificador convolucional, el segundo *bit* de paridad son todos los *bits* de las posiciones impares de la salida del segundo codificador convolucional y finalmente los últimos 12 *bits* se agregan de los *bits* de Trellis en la salida de cada codificador convolucional.

Al observar los resultados obtenidos de cada codificador convolucional mediante los comandos en MATLAB, y comparar con los resultados obtenidos en el software implementado, son los mismos, validando de esta manera el proceso desarrollado.

5.4.4. Adaptación de tasa

Este proceso no se encuentra en la versión que se ha utilizado de MATLAB, por esta razón para validar su funcionamiento se utiliza un código realizado el libro “LTE WITH MATLAB”, y se comparan los resultados con el software implementado.

En las Figuras 5.11 y 5.12 se presentan los resultados obtenidos con el software implementado y con MATLAB, en donde se visualiza que los resultados varían, esto se debe a que el código del libro emplea un método diferente para el cálculo del *buffer* circular, y la normativa 3GPP posee dos opciones; en el software está desarrollado el método indicado para el canal de datos y en el cual intervienen variables como: versión de redundancia, el equipo de usuario, el tipo de modulación, entre otras.

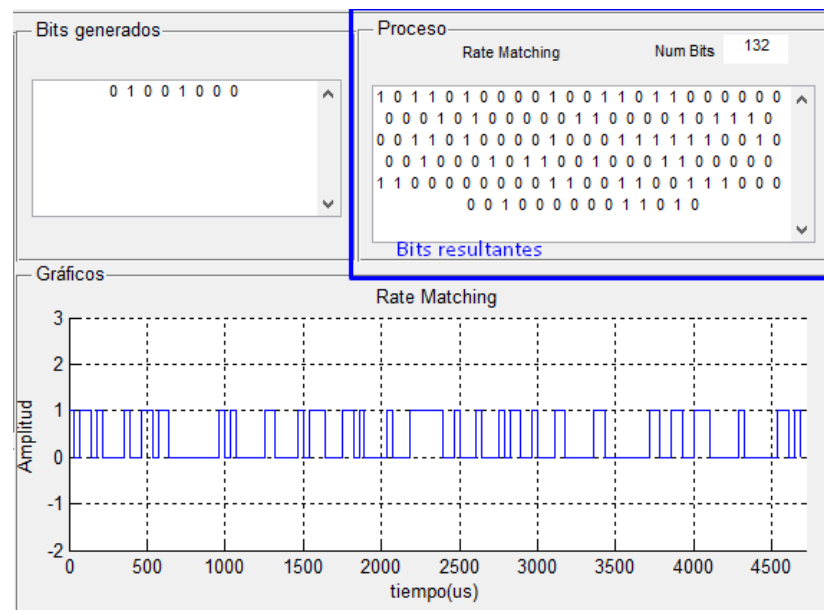


Figura 5.11: Adaptación de tasa, software implementado.

```
>> Kplus=50;  
>> Rate=1/3;  
>> in=[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 0 1 0 0 1 0 1 1 0 1 0 1  
>> y=RateMatcher(in, Kplus, Rate);  
>> y'
```

función libro "LTE with Matlab"

```
ans =
```

Columns 1 through 24

1	0	1	0	1	1	0	1	0	0	0	0	1	0	0	1	1	0	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Columns 25 through 48

0	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Columns 49 through 72

0	1	0	0	1	0	1	1	0	0	0	1	0	1	0	0	1	0	1	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Columns 73 through 96

0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Columns 97 through 120

1	0	0	0	0	0	0	1	0	0	1	1	0	1	1	1	0	0	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Columns 121 through 132

0	0	0	1	0	1	0	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---

Figura 5.12: Adaptación de tasa, MATLAB.

5.4.5. Aleatorización

Los bits que ingresan a este proceso de aleatorización, son la unión de los códigos de bloque de cada adaptación de tasa (*rate match*), para el ejemplo son:
 1 0 1 1 0 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1
 0 1 1 1 0 0 0 1 1 0 1 0 0 0 0 1 0 0 0 1 1 1 1 1 1 0 0 1 0 0 0 1 0 0 0 1 0 1 1 0 0 1
 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0
 0 0 1 1 0 1 0. El resultado de este proceso se determina mediante una operación XOR entre los bits de entrada y los bits obtenidos por una secuencia *gold*. En la Figura 5.13 se presenta este proceso ejecutado en el software implementado.

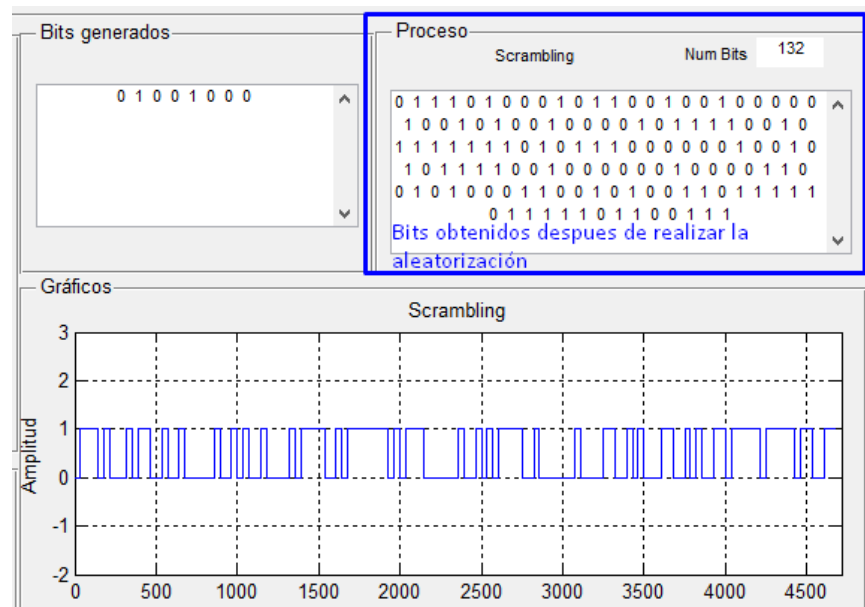
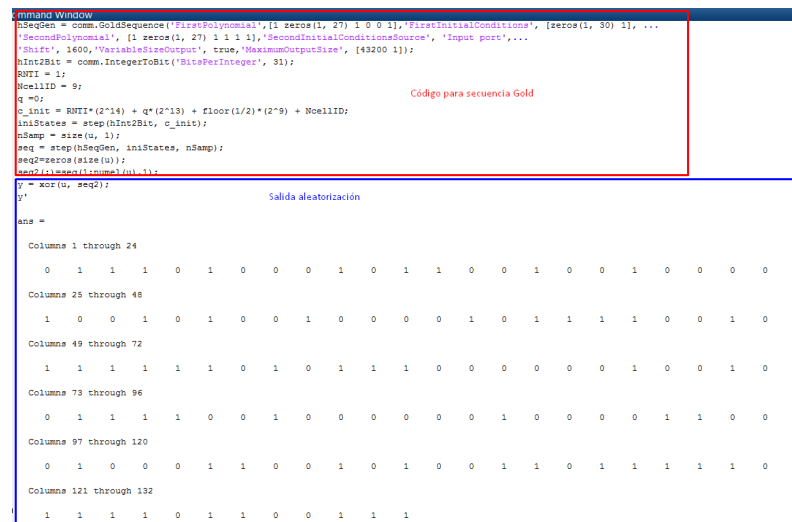


Figura 5.13: Aleatorización, software implementado.

Para la validación en MATLAB, la Figura 5.14, muestra el proceso para generar la secuencia *gold*, y el comando para generar la aleatorización. Al comparar las Figuras 5.13 y 5.14, los resultados obtenidos son los mismos, validando de esta manera el proceso desarrollado.



```

Command Window
hSeqGen = comm.GoldSequence('FirstPolynomial',[1 zeros(1, 27) 1 0 0 1], 'FirstInitialConditions', [zeros(1, 30) 1], ...
'SecondPolynomial',[1 zeros(1, 27) 1 1 1 1], 'SecondInitialConditionsSource', 'Input gold',...
'Shift', 1600, 'VariableSizeOutput', true, 'MaximumOutputSize', [43200 1]);
hInt2Bit = comm.IntegerToBit('BitsPerInteger', 31);
RNTI = 1;
MoellID = 9;
q = 0;
c_init = RNTI*(2^14) + q*(2^13) + floor((1/2)*(2^9) + MoellID);
iniStates = step(hInt2Bit, c_init);
nSamp = size(u, 1);
seq = step(hSeqGen, iniStates, nSamp);
seq2=zeros(size(u));
seq2(1:mod(nSamp,16)+1:16:nSamp)=seq(1:16:nSamp);

r = xor(u, seq2);
ans =
Columns 1 through 24
0 1 1 1 1 0 1 0 0 0 1 0 1 1 0 0 1 0 0 1 0 0 0 0
Columns 25 through 48
1 0 0 1 0 1 0 0 1 0 0 0 0 1 0 1 1 1 1 0 0 1 0 1
Columns 49 through 72
1 1 1 1 1 1 0 1 0 1 1 1 0 0 0 0 0 0 1 0 0 1 0 1
Columns 73 through 96
0 1 1 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0
Columns 97 through 120
0 1 0 0 0 1 1 0 0 1 0 1 0 0 1 1 0 1 1 1 1 1 0 1
Columns 121 through 132
1 1 1 1 0 1 1 0 0 1 1 1

```

Figura 5.14: Aleatorización, MATLAB.

5.4.6. Modulación

La tecnología LTE presenta sus propios esquemas de modulación, en donde los *bits* de entrada a este proceso son los obtenidos en la aleatorización. A continuación se muestra una modulación QPSK realizado mediante el software implementado y mediante el comando `comm.PSKModulator` en de MATLAB configurado como lo indica la norma 3GPP. Al observar las figuras 5.15, 5.16 y 5.17, los resultados obtenidos en cada caso son iguales, validando de esta manera el proceso desarrollado.

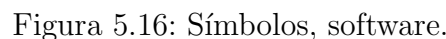


Figura 5.17: Modulaci3n y s3mbolos, MATLAB.

5.4.7. Demodulación

La entrada de este proceso son los símbolos recibidos mediante OFDM y MI-MO, además se desarrollaron los esquemas indicados en la 3GPP, a continuación en las Figuras 5.18 y 5.19 se muestran los *bits* obtenidos en el software implementado y en MATLAB respectivamente.

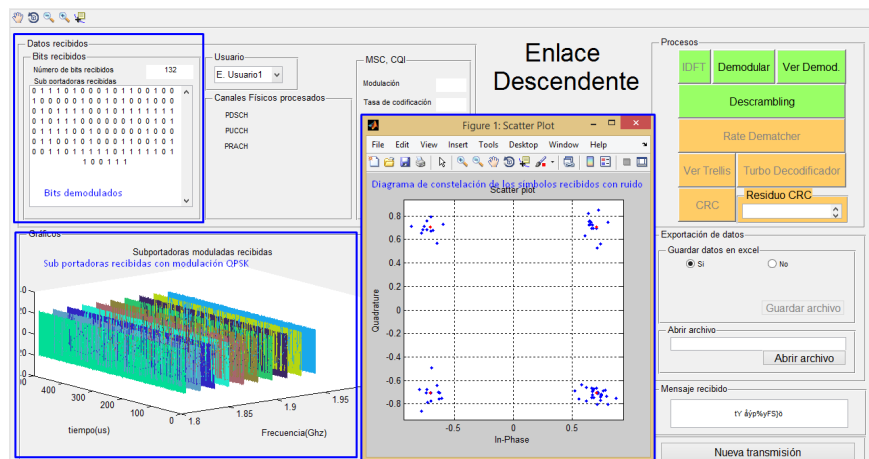


Figura 5.18: Demodulación, software.

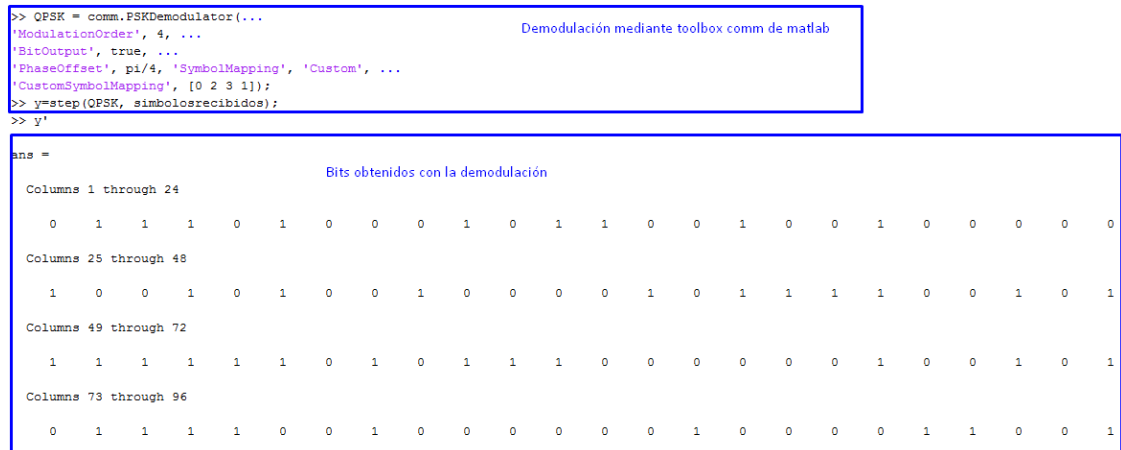


Figura 5.19: Demodulación, MATLAB.

5.4.8. Des-aleatorización

Este proceso es idéntico al de aleatorización, puesto que la secuencia *gold* es un código ortogonal; al realizar un XOR entre los *bits* recibidos y esta secuencia, se obtiene el tren de *bits* original previo a la aleatorización. En las Figuras 5.20 y 5.21, se muestra el resultado de este proceso, mediante el software implementado y mediante MATLAB, validando de esta manera el software implementado.

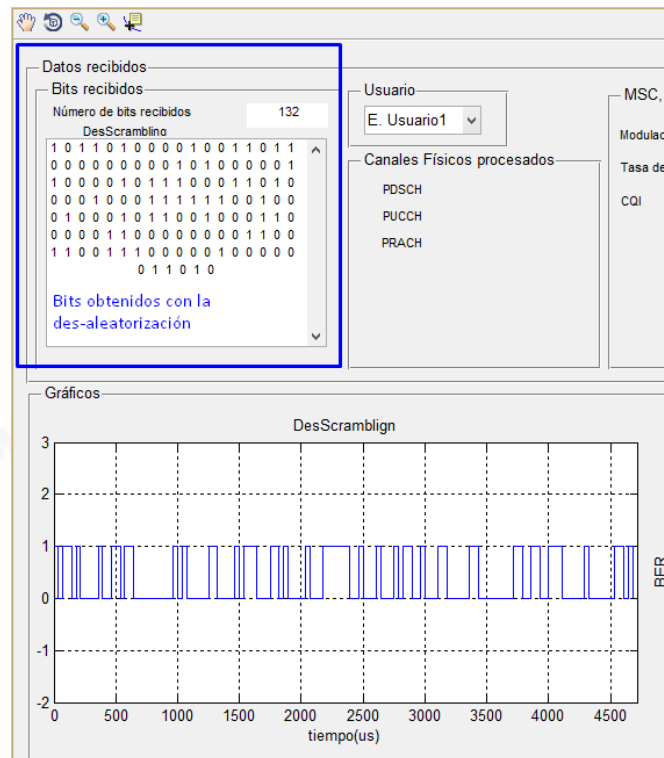


Figura 5.20: Des-aleatorización, software.

```
>> u=[0 1 1 1 0 1 0 0 0 1 0 1 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 0 1 0 1 1 1 1 0 0 1 0
>>
maxG=43200;
hSeqGen = comm.GoldSequence('FirstPolynomial',[1 zeros(1, 27) 1 0 0 1],...
'FirstInitialConditions', [zeros(1, 30) 1], ...
'SecondPolynomial', [1 zeros(1, 27) 1 1 1 1],...
'SecondInitialConditionsSource', 'Input port',...
'Shift', 1600,...
'VariableSizeOutput', true,...
'MaximumOutputSize', [maxG 1]);
hInt2Bit = comm.IntegerToBit('BitsPerInteger', 31);
RNTI = 1;
NcellID = 9;
q = 0;
c_init = RNTI*(2^14) + q*(2^13) + floor(1/2)*(2^9) + NcellID;
iniStates = step(hInt2Bit, c_init);
nSamp = size(u, 1);
seq = step(hSeqGen, iniStates, nSamp);
seq2=zeros(size(u));
seq2(:)=seq(1:numel(u),1);
y = xor(u, seq2);
>> y'
```

Proceso de des-aleatorización mediante el uso de la secuencia gold del toolbox de comunicaciones de MATLAB.

Resultado de la des-aleatorización.

```
ans =

Columns 1 through 24
    1     0     1     1     0     1     0     0     0     0     1     0     0     1     1     0     1     1     0     0     0     0     0     0

Columns 25 through 48
    0     0     0     1     0     1     0     0     0     0     0     0     1     1     0     0     0     0     1     0     1     1     1     0     0

Columns 49 through 72
    0     1     1     0     1     0     0     0     0     0     1     0     0     0     1     1     1     1     1     1     0     0     1     0     0

Columns 73 through 96
    0     1     0     0     0     1     0     1     1     0     0     1     0     0     0     1     1     0     0     0     0     0     0     1     1

Columns 97 through 120
    0     0     0     0     0     0     0     0     1     1     0     0     1     1     0     0     1     1     1     0     0     0     0     0     0

Columns 121 through 132
    1     0     0     0     0     0     0     0     1     1     0     1     0
```

Figura 5.21: Des-aleatorización, MATLAB.

5.4.9. Decodificación

Para el proceso de la decodificación se realiza mediante la codificación de Trellis, la misma que está implementada en el software desarrollado. En las Figuras 5.22 y 5.23, se visualiza este proceso mediante el software implementado y mediante MATLAB, en donde, los resultados varían en ciertos *bits*, esto se presenta debido al método que usa MATLAB internamente para la decodificación.



Para el caso de la recepción, el residuo del proceso CRC es igual a cero, para comprobar que el mensaje llega sin ningún error;0 el proceso de CRC se realiza de igual manera que en el envío, puesto que es un código ortogonal y como *bits* de entrada a este proceso se utilizan los *bits* obtenidos en el proceso de decodificación. En las Figuras 5.24 y 5.25, se presenta la validación del software implementado, comparando los resultados obtenidos, con el módulo de MATLAB.

Como se puede observar, en ambos casos los *bits* obtenidos son los mismos, y el residuo del CRC es cero por lo que se comprueba que no hay errores en el tren de *bits* recibidos así como en la decodificación previa.

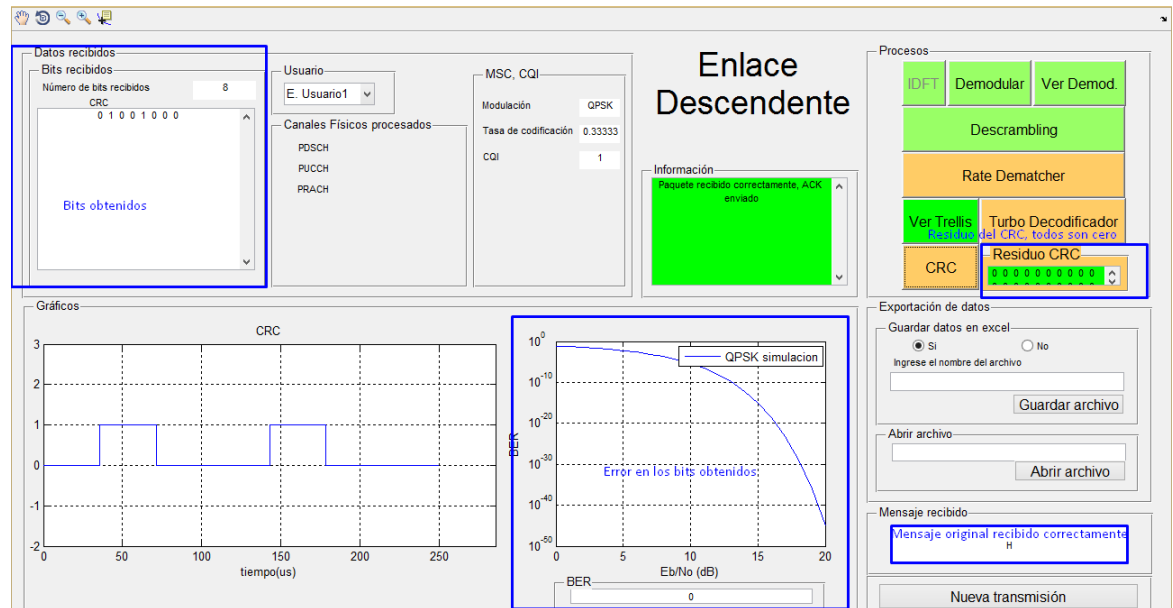


Figura 5.24: Proceso CRC recepción, software.

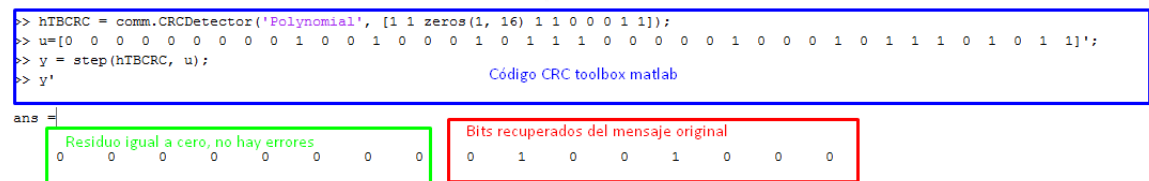


Figura 5.25: Proceso CRC recepción, MATLAB.

5.5. Simulaciones realizadas

Dentro de esta sección se presentan las gráficas OFDM y MIMO, espectro de frecuencias, BER; obtenidos de distintas simulaciones realizadas bajo diferentes parámetros mediante el software implementado.

En la primera simulación abarca la mayoría de datos que nos permite analizar el software, en las siguientes simulaciones se presenta las gráficas MIMO, OFDM, diagrama de constelación y el BER final.

5.5.1. Primera simulación

Datos configurados dentro de la primera simulación.

- Modulación: QPSK
- Adaptación de tasa: 1/3
- EbNo: 5 dB
- Potencia: 20 W
- Número de bits generados: 128
- Desincronización: No
- Configuración MIMO: Redundancia 2x2
- ID Celda: 9
- Número de usuarios: 1

Resultados obtenidos.

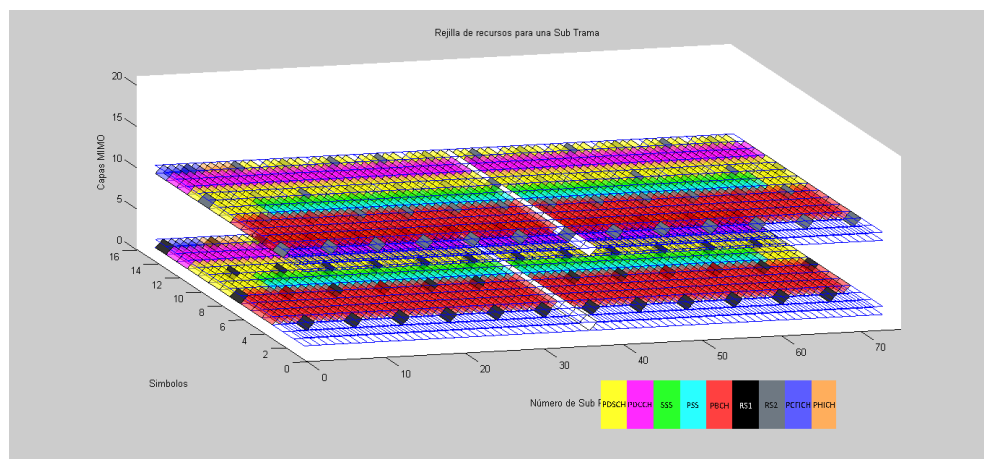


Figura 5.26: Capas MIMO.

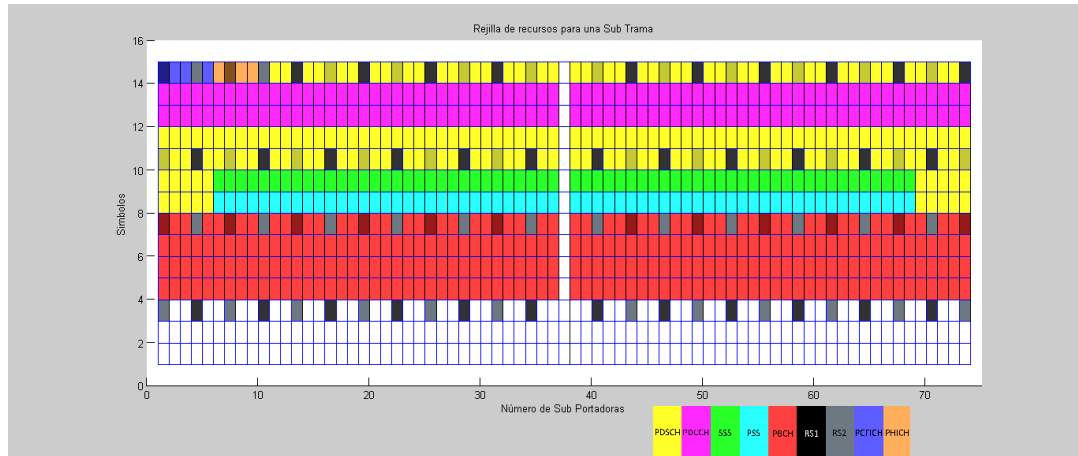


Figura 5.27: Rejilla OFDM.

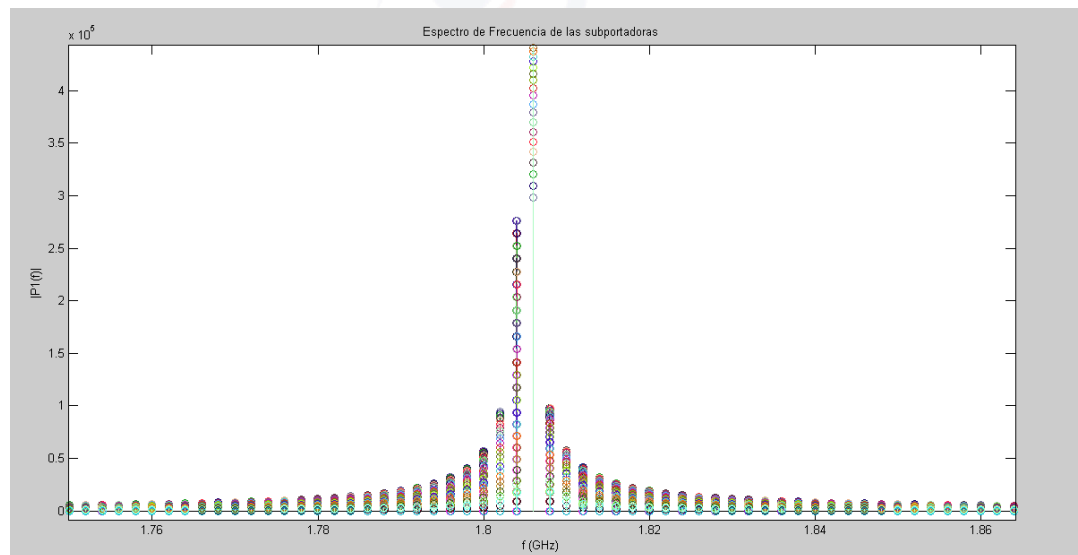


Figura 5.28: Espectro de frecuencia.

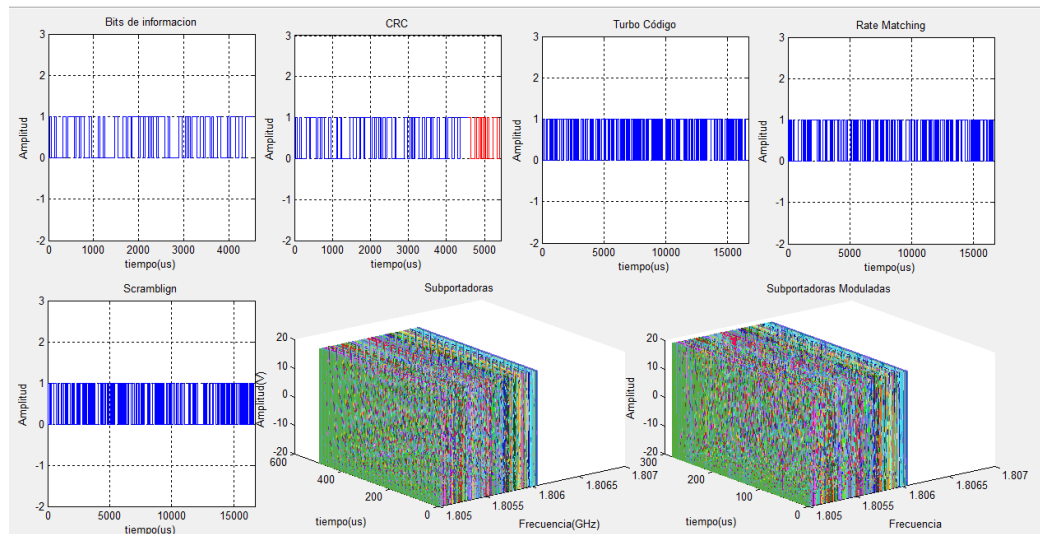


Figura 5.29: Procesos y subportadoras.

Salida de datos	
Datos	
Número de bits generados	128
Número de bits por símbolo	2
Frecuencia Inicial (GHz)	1.805
Tiempo de bit (us)	35.7143
Bloque de código (bits)	152
Bloque de transporte (bits)	152
Capacidad bruta (Mb/s)	4.032
Tasa PDSCH (Mb/s)	0.936

Figura 5.30: Datos de salida.

MIMO/OFDM

Datos

Bits canal de información

```
1 0 0 0 1 1 0 0 1 1 1 1 1 0 1 1 1 1 0 0 0 0 1 1 0 0
```

Bits Serie-Paralelo

```
1 0 0 0 1 1 0 0 1 1 1 1 1 0 1 1 1 1 0 0 0 0 1 1 0 0
0 0 0 0 1 0 1
1 1 1 0 0 1 0 0 1 1 0 0 1 1 1 1 0 0 0 1 0 1 1 0 0 1
0 0 0 0 0 1 0
0 1 0 0 1 0 1 1 1 1 0 1 0 1 1 0 1 0 1 1 1 0 0 1 0
1 0 0 1 0 1 1
0 1 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0 1 0 0 1 0 1 1 1
0 0 1 1 1 1 0
1 0 1 0 0 0 1 0 0 1 0 0 1 0 0 1 1 1 1 1 1 0 0 1 0 1
0 1 1 0 0 1 1
0 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 0 0 0 0 1 1 1 0
```

Datos OFDM

Número de ER ocupados por PDSCH	238
Número de Slots	1
Número de sub tramas	1
Número de Tramas	1
Número de bloques de recurso utilizados	3
Tiempo Slot (ms)	0.5
Tiempo Trama (ms)	10
Número de sub portadoras	72
Ancho de Banda Normalizado (MHz)	1.4
Número de bits por ER (Kb/s)	28
Número de bits por BR (Kb/s)	336

Iniciar

Ayuda

Cerrar

Datos MIMO

Configuración: SU-MIMO

Número de puertos de antenas Tx: 2

Número de capas MIMO: 1

Figura 5.31: Datos MIMO/OFDM.

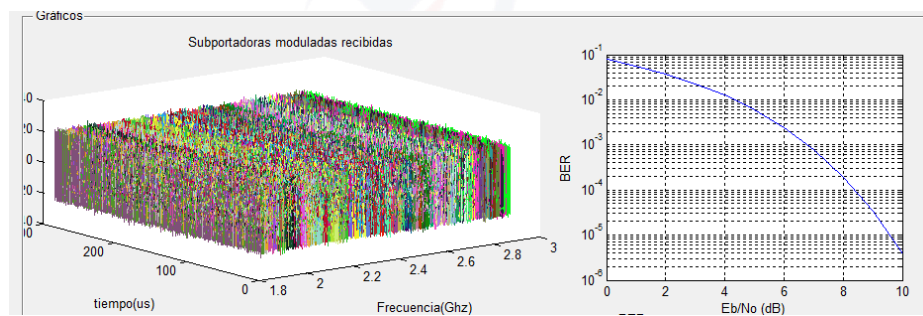


Figura 5.32: Subportadoras recibidas con ruido, BER=0.

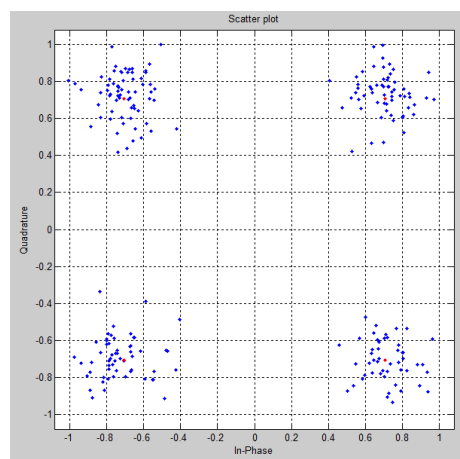


Figura 5.33: Diagrama de constelación.

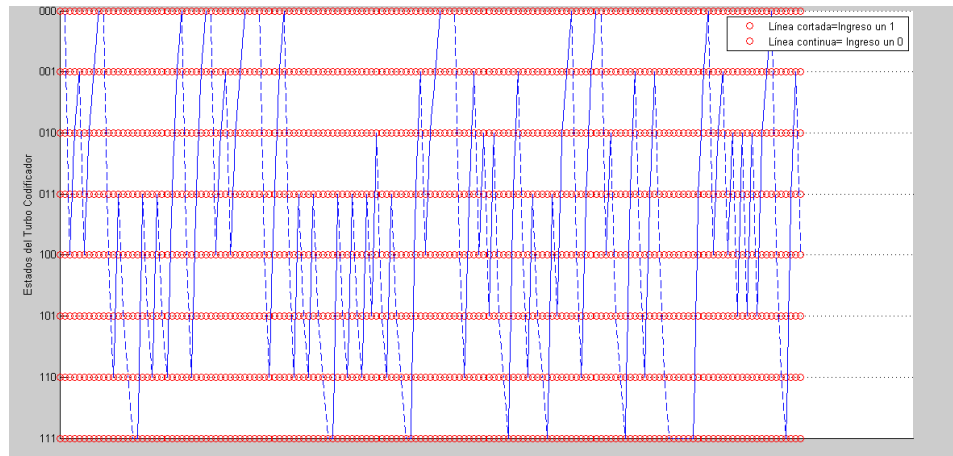


Figura 5.34: Diagrama de Trellis.

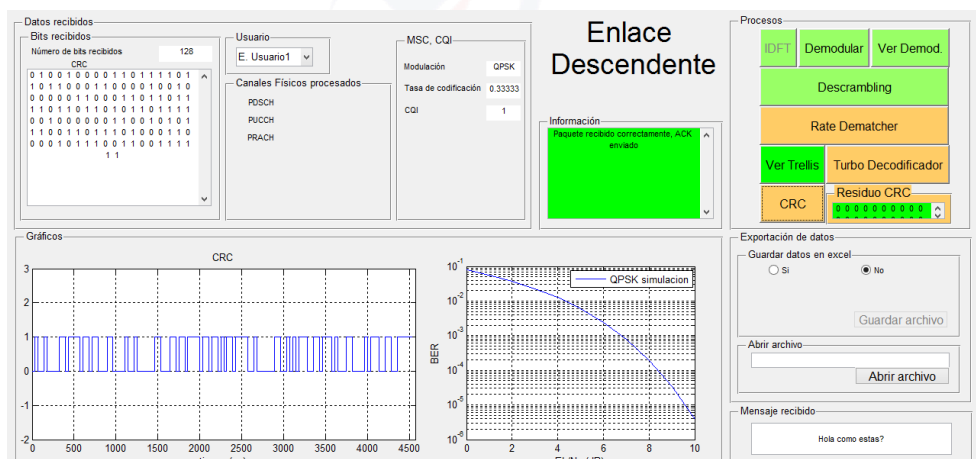


Figura 5.35: Datos recibidos, BER=0, CQI, ACK.

5.5.2. Segunda simulación

En esta simulación se muestra el caso de dos usuarios con diferentes modulaciones. Datos configurados dentro de la segunda simulación.

- Modulación: 64QAM usuario1, 16QAM usuario2
- Adaptación de tasa: 1/3
- EbNo: 15 dB
- Potencia: 20 W
- Número de bits generados: 96 para cada usuario
- Desincronización: No
- Configuración MIMO: Redundancia 2x2
- ID Celda: 8
- Número de usuarios: 2

Resultados obtenidos.

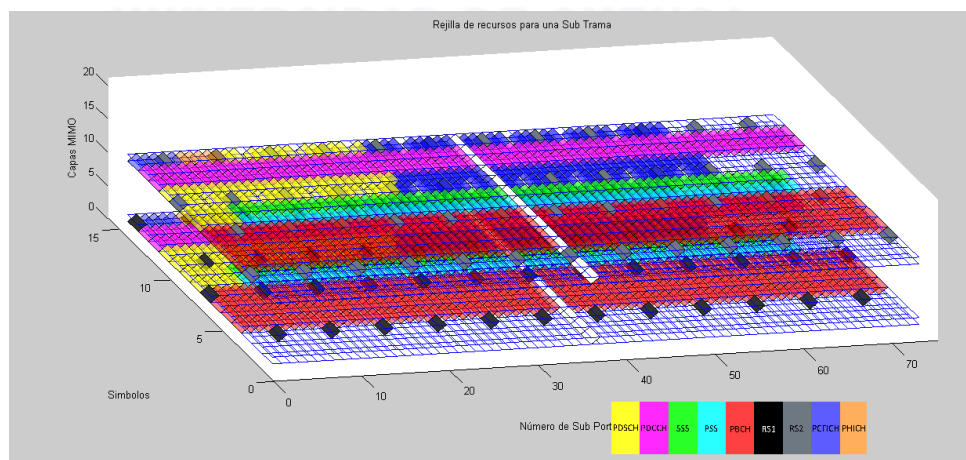


Figura 5.36: Capas MIMO.

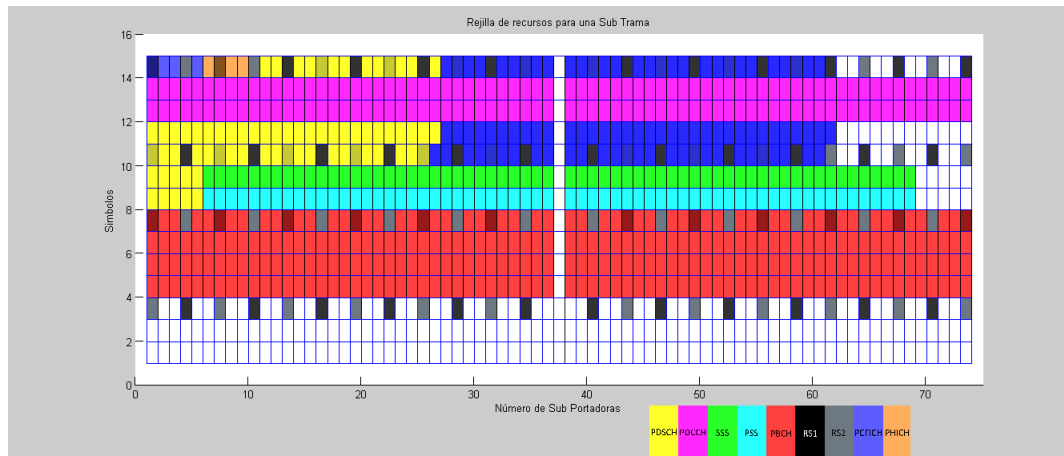


Figura 5.37: Rejilla OFDM.

Mediciones en la recepción del usuario 1.

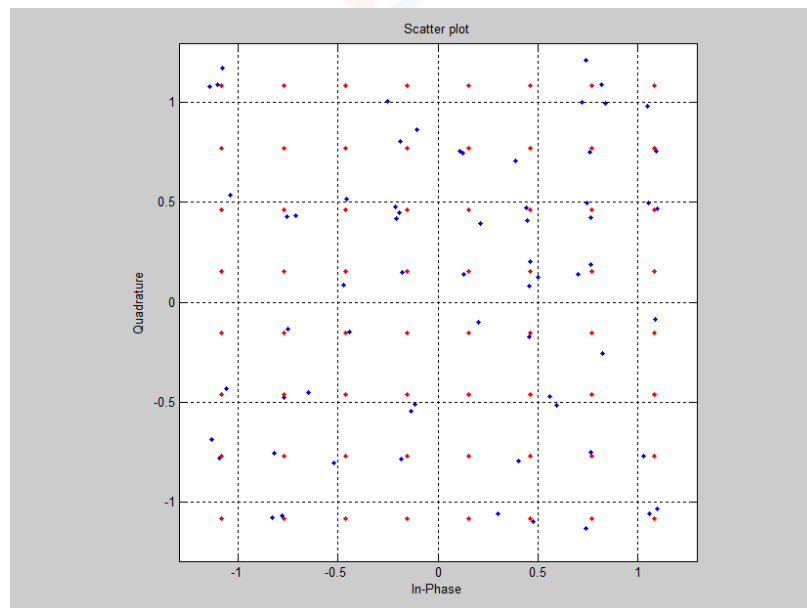


Figura 5.38: Diagrama de constelación recibido.

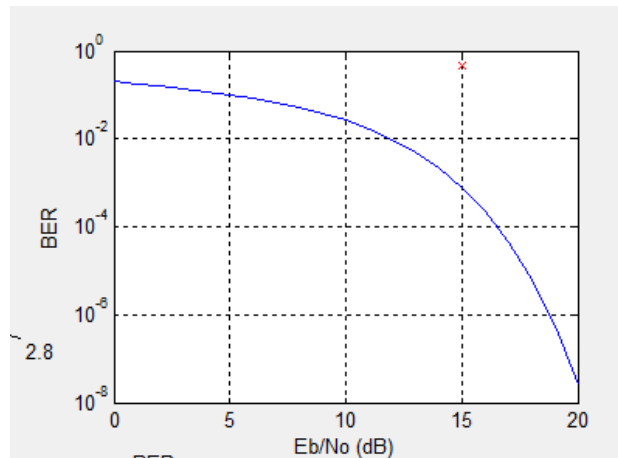


Figura 5.39: BER información recibido.

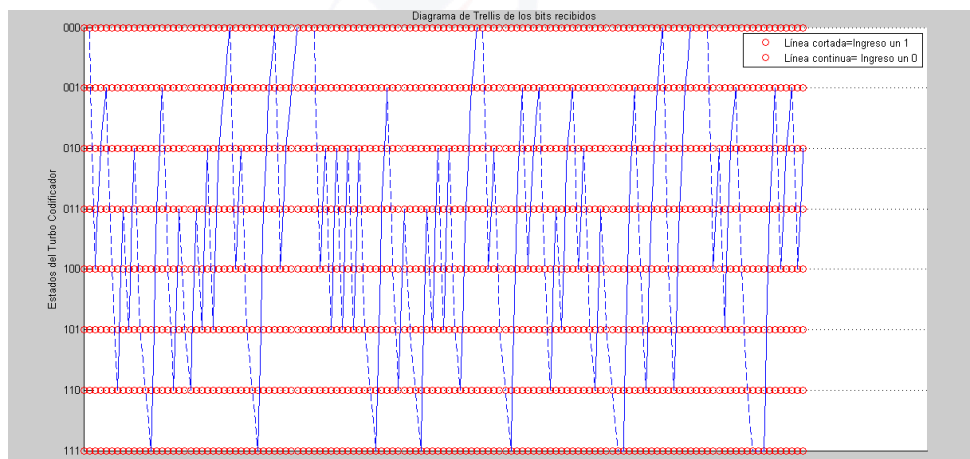


Figura 5.40: Diagrama de Trellis.

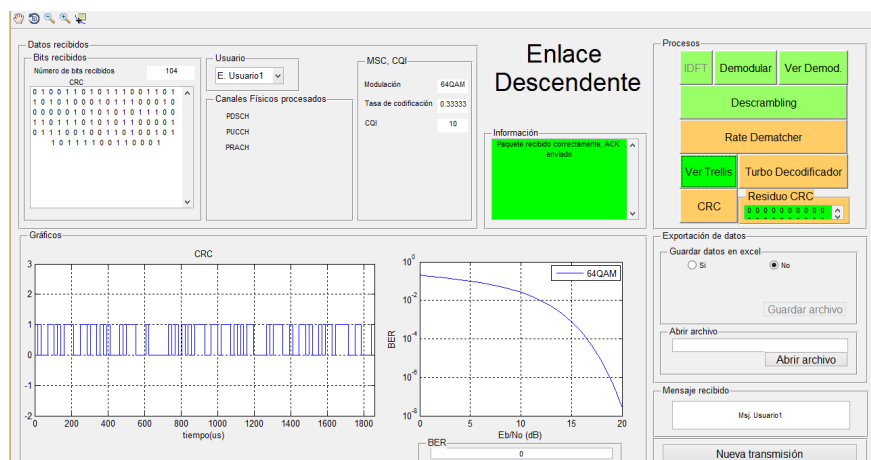


Figura 5.41: BER con corrección de errores.

Mediciones en la recepción del usuario 2.

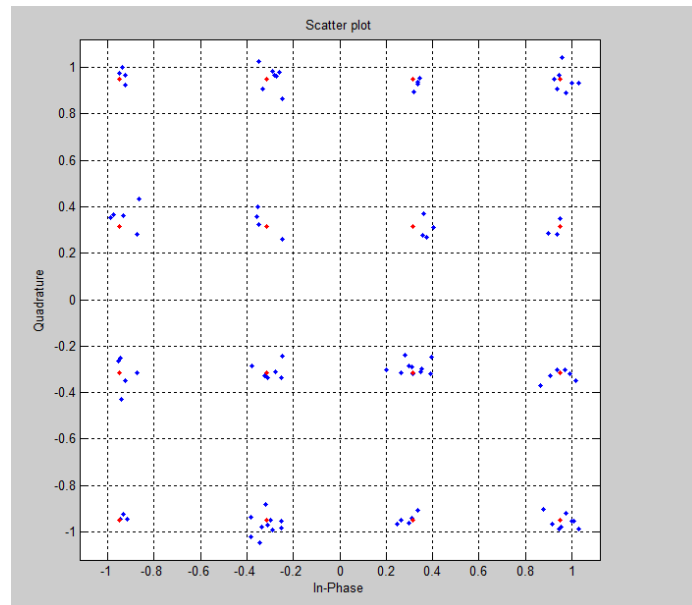


Figura 5.42: Diagrama de constelación.

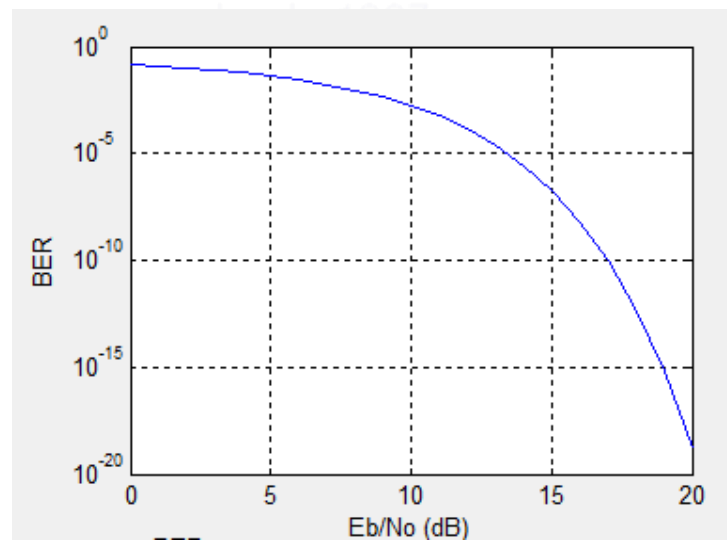


Figura 5.43: BER información recibido.

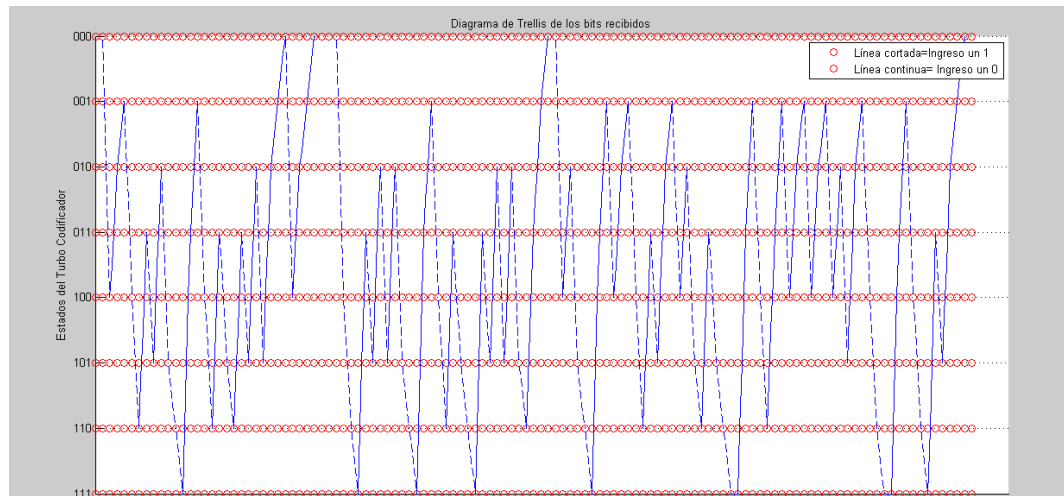


Figura 5.44: Diagrama de Trellis.

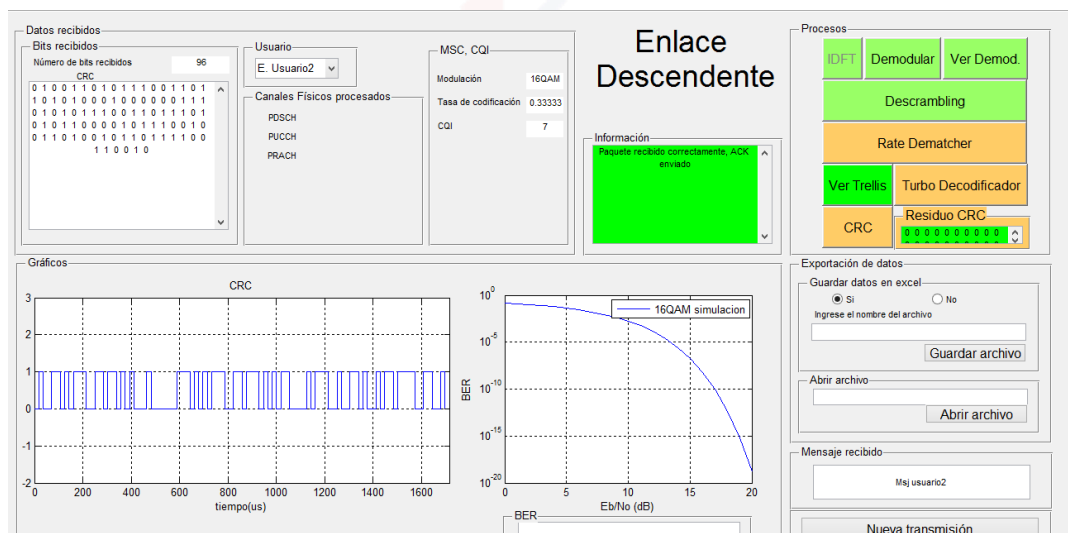


Figura 5.45: BER con corrección de errores.

5.5.3. Tercera simulación

En esta simulación se realiza un proceso de envío de información en el cual se compara el BER recibido, solo se varía el esquema de modulación, el resto de datos se mantienen iguales.

- Adaptación de tasa: 1/3
- EbNo: 5 dB
- Potencia: 15 W
- Número de bits generados: 128
- Desincronización: no
- Configuración MIMO: Redundancia 2x2
- ID Celda: 8
- Número de usuarios: 1

Resultados obtenidos.

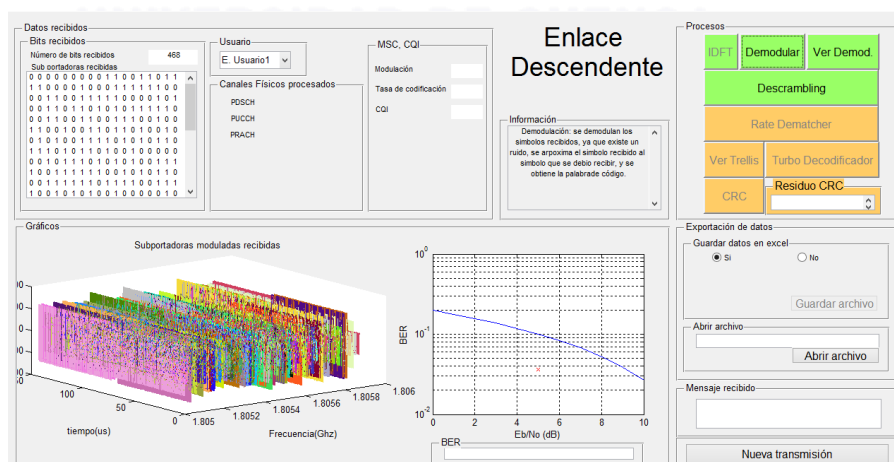


Figura 5.46: Modulación 64QAM.

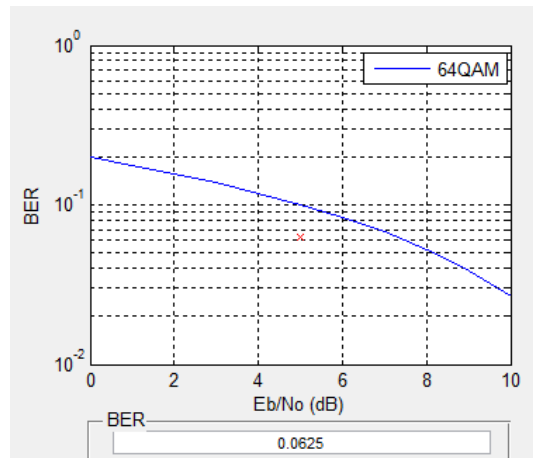


Figura 5.47: BER con corrección de errores.

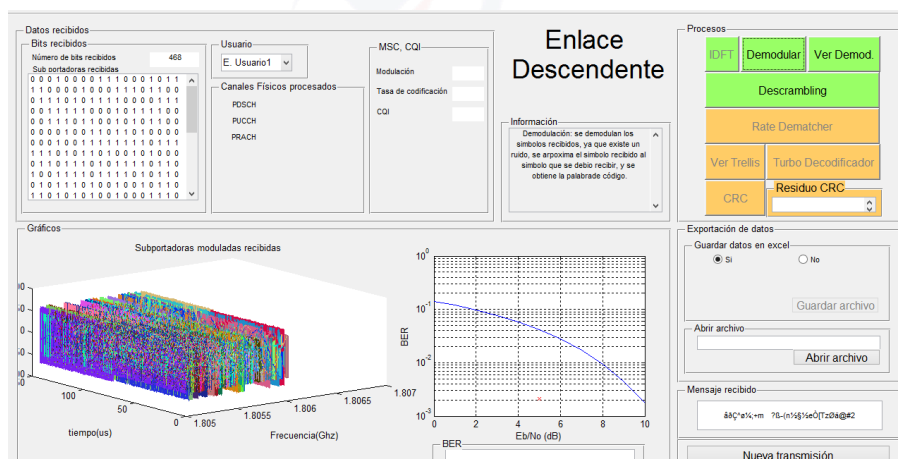


Figura 5.48: Modulación 16QAM.

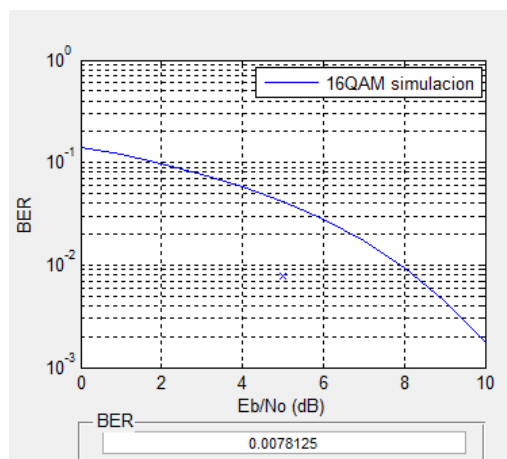


Figura 5.49: BER con corrección de errores.

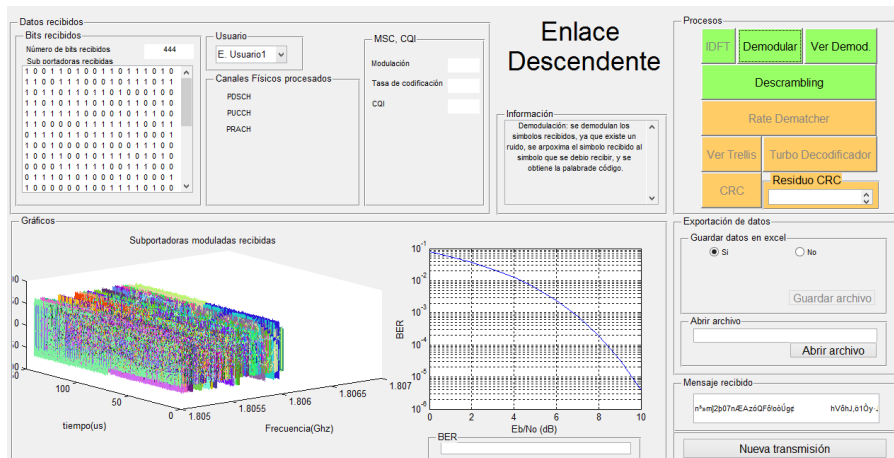


Figura 5.50: Modulación QPSK.

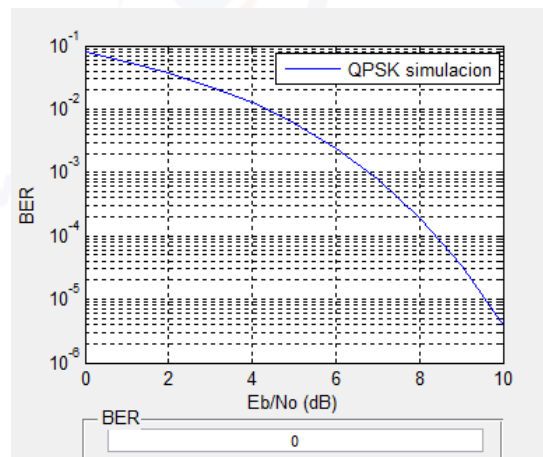


Figura 5.51: BER con corrección de errores.

Capítulo 6

Conclusiones y Recomendaciones.

6.1. Conclusiones

La implementación de esta herramienta de simulación presentó un gran reto, debido a la gran cantidad de especificaciones y normas existentes en relación a la tecnología LTE y su capa física principalmente, las cuales muchas veces presentan los procesos desde un punto de vista teórico y no necesariamente basados en la implementación práctica. Además de esto, el hecho de realizar una interfaz didáctica representó un reto adicional, ya que se tuvo que poner gran énfasis en la parte gráfica y de diseño del software para que los procesos puedan ser claramente entendidos y analizados. Esto nos llevó a la tarea de implementar cada uno de los algoritmos y procesos con nuestro propio código, entendiendo la tecnología a detalle y desde el procesamiento de cada bit, para así poder cumplir con el objetivo planteado, ya que de haberse usado módulos, o comandos propios de la plataforma MATLAB, no hubiese sido posible mostrar el detalle de cada proceso y por ende la herramienta no hubiese resultado didáctica, recordando que esta herramienta busca ser un complemento a la teoría y la cual está dirigida para el uso de estudiantes.

Cabe recalcar que las especificaciones 3GPP, tienen un enfoque bastante teórico, de los procesos y de cada una de las etapas y capas de LTE, lo cual nos llevó a encontrarnos con ciertas dificultades a la hora de la implementación de dichos procesos y a recurrir a otro tipo de bibliografía en la cual precisamente se presentaba un enfoque mucho más práctico, a partir de la experiencia de la implementación de la teoría en la práctica. De esta manera, el haber realizado cada uno de los algoritmos no solo nos ayudó a comprender a profundidad la tecnología, sino que nos ayudó a diseñar de mejor forma el programa, ya que nos permitió controlar



CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES.

cada una de las variables tanto de salida como de entrada de estos algoritmos, además que se pudo contrastar con los módulos de MATLAB, y en todos los casos, los algoritmos implementados trabajaron con la misma eficiencia.

Una de las desventajas de haber implementado una herramienta muy didáctica y con gran cantidad de gráficos, y dado el alto número de datos procesados en las simulaciones, es la velocidad de procesamiento, por lo que se optó por no mostrar ciertos gráficos como los de las rejillas OFDMA y SC-FDMA cuando se procesen más de mil bits. Así mismo se vio que resultaba innecesario ver los procesos paso a paso para una cantidad tan grande de bits, por lo que estas opciones están disponibles para procesos que comprendan una menor cantidad de bits. Debido a todo esto se decidió implementar adicionalmente en el software un simulador automático, es decir, que realice todos los procesos internamente y ejecute una comunicación de enlace ascendente y descendente, sin la necesidad de realizarlo paso a paso, y en donde se puede observar de igual manera datos como el BER, y así tener también como opción una simulación mucho más rápida sin tanto contenido gráfico.

En cuanto a los procesos tanto de enlace ascendente como descendente, se ha podido implementar de forma eficiente cada uno de ellos, representando un gran reto dichas implementaciones.

Uno de los procesos más abstractos y difíciles de mostrar de forma didáctica fue las técnicas MIMO, las mismas que presentan un procesamiento mayormente físico, y difícil de mostrar en un simulador; sin embargo, se ha podido representar e implementar este proceso de forma adecuada para los fines pertinentes.

Si bien todos los procesos implementados representaron gran dificultad, el proceso que mayor reto presentó fue el de decodificación, ya que en las normativas 3GPP no se indica a detalle cómo realizar el mismo, simplemente se explica que éste funciona mediante el diagrama de Trellis.

Después de una amplia investigación al respecto, se pudo verificar que no existe un algoritmo específico desarrollado para Trellis, y que la eficiencia en la decodificación podría variar de acuerdo a como se lo haya implementado, en la mayoría de casos en los que se necesitaba utilizar este tipo de decodificación se



CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES.

usaba el módulo propio de MATLAB, lo cual representaba un inconveniente ya que como se indicó anteriormente, nuestro software plantea la explicación paso a paso de cada proceso.

De esta manera se procedió a la implementación de este algoritmo, el mismo que nos dio resultados muy buenos, pudiendo no solo corregir palabras de código con errores altos, sino que se pudo mostrar de forma gráfica el diagrama de Trellis para cada simulación, haciendo de este uno de los procesos mejor explicados didácticamente dentro del simulador. Este algoritmo tiene una eficiencia del 100 por ciento de corrección de errores, si el error no es mayor al 50 por ciento dentro de los bits sistemáticos, de paridad 1 y paridad 2.

En el software prácticamente todo los procesos pueden ser vistos paso a paso, salvo en la recepción, en donde se optó por no mostrar paso a paso tanto el “*DeScrambling*” como el “*Rate Dematcher*”, ya que sus procesos son similares a los realizados en el envío de los datos.

En el caso del “*DeScrambling*” al ser un código ortogonal, se recupera la información al multiplicar la palabra de código recibida por el mismo código *Scrambler*, y en el caso del “*Rate Dematcher*”, es el proceso directamente inverso al realizado en el envío.

Uno de los objetivos de este proyecto era realizar una validación de las simulaciones realizadas con nuestro software, contrastándolas con mediciones reales realizadas en la ciudad de Cuenca, sin embargo al solicitar esta información a la ARCOTEL, se nos indicó que aún no se disponen de estas mediciones en la ciudad. Por lo tanto se ha analizado la validez del software a partir de una prueba de escritorio, la misma que se muestra como ejemplo ilustrativo en el Capítulo 4. Aquí se realiza todo el proceso de forma manual, y se compara con los resultados obtenidos en el software, en donde como se puede ver, estos resultados coinciden en cada uno de los procesos realizados, dando así validez a los algoritmos implementados para cada uno de los procesos.

Además se han realizado pruebas ingresando diferentes valores, los mismo que se han comparado con tablas como las del CQI establecido por la normativa 3GPP, en donde se obtienen resultados similares a los indicados en dicha tabla.



Uno de los puntos a futuro sería realizar esta validación con datos reales; sin embargo, cabe recalcar, que nuestro simulador funciona en un ambiente “ideal”, por lo que posiblemente varíen con respecto a las mediciones reales, en donde a su vez se presentan gran variación entre las mediciones, ya que en las mismas intervienen muchos parámetros o variables del entorno.

6.2. Recomendaciones

Después del análisis de una tecnología de suma importancia en las telecomunicaciones actuales como es LTE, enfocándonos en su capa física, y del desarrollo de un software que modela cada uno de sus procesos, y a la vez muestra de forma didáctica, se plantean las siguientes recomendaciones:

- Si bien la plataforma de MATLAB es muy buena para el trabajo con matrices y muy eficiente para ejecutar los algoritmos implementados en cuanto a codificación, modulación, entre otros, y de disponer de un entorno de desarrollo gráfico como lo es el GUI, se puede buscar una opción que permita implementar este tipo de entornos con alto contenido gráfico y a la vez didáctico, como el realizado, pero sin tener tanta demora en el procesamiento de los datos y de la parte gráfica.
- El modelar o programar, en funciones o algoritmos, procesos de este tipo de tecnologías de telecomunicaciones, en las cuales a nivel de capa física encontramos sus características fundamentales para la comunicación, requiere además de un fundamento teórico, un análisis de escritorio para garantizar el correcto funcionamiento de las mismas, ya que así se evidencian posibles errores al momento de la implementación en el software, esto con el fin de garantizar que el software sea una correcta implementación de los procesos reales, que en algunos casos, muchos de ellos se realizan de forma analógica, o mediante otros métodos y en entornos diferentes al de un software.
- Mantener un equilibrio entre el funcionamiento correcto de los procesos, y la forma didáctica en la que estos se muestran, es decir, implementar algoritmos que garanticen el correcto funcionamiento del programa, y estos nos permitan una clara visualización de cada proceso y así entender de una mejor manera los mismos.



CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES.

- Utilizar este software en las universidades para una enseñanza más didáctica, puesto que ayuda al estudiante a tener un mejor aprendizaje, poniendo en práctica los conceptos estudiados teóricamente.



Apéndice A

Manual de Usuario

A.1. Menú Inicial

1. Ejecutar en MATLAB el script con el nombre “PrincipalTesis”, y se abre la siguiente ventana.

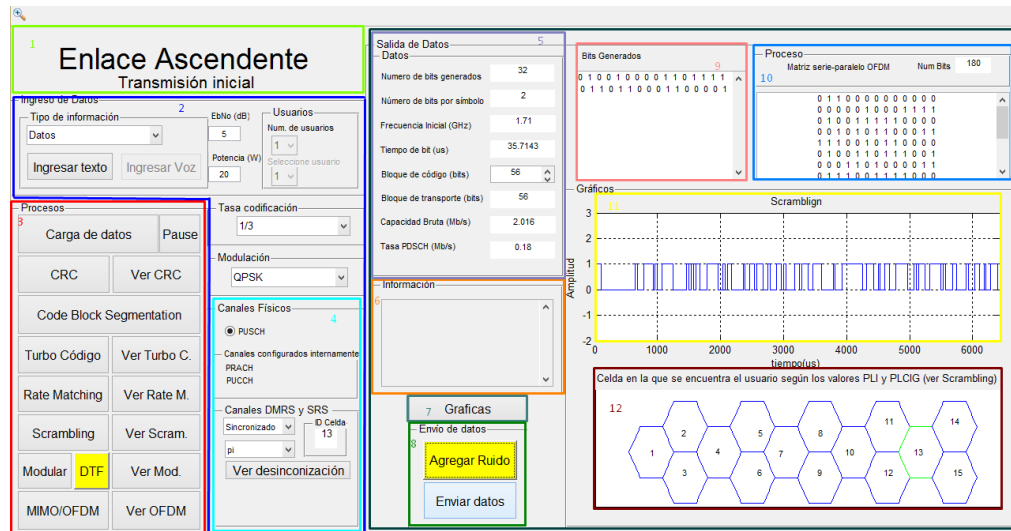


Aquí muestra el menú principal, el mismo que consta de 3 opciones: “Enlace ascendente”, “Enlace descendente” y “Enlace con procesos automáticos”

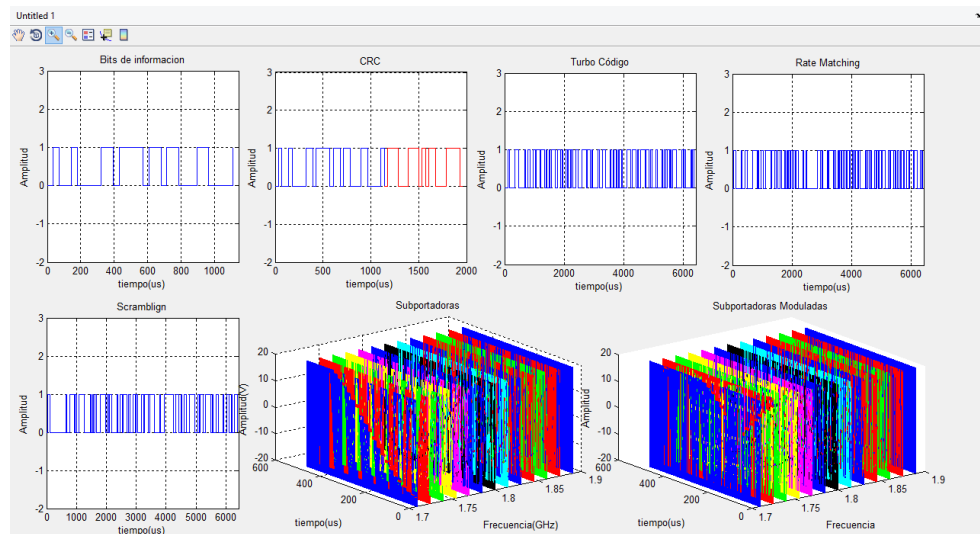
2. Seleccionar la opción que se desee ejecutar.

A.2. Estructura de ventanas

A.2.1. Ventana de envío

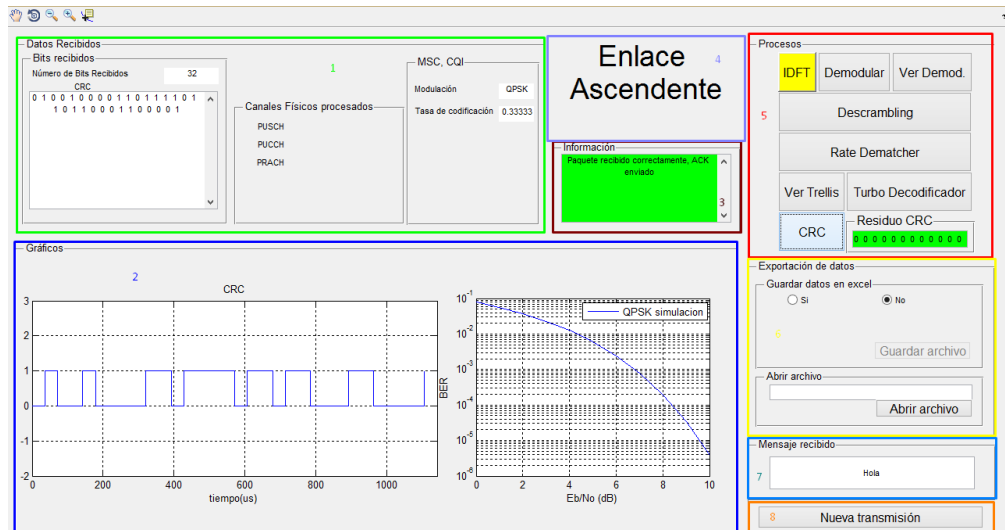


1. Título, indica el tipo de enlace y el número de transmisión.
2. Panel de ingreso de datos y configuración de parámetros de entrada.
3. Panel de procesos de la capa física.
4. Panel de ingreso de datos, aquí se configuran ciertos parámetros de los canales físicos.
5. El panel de la salida de datos muestra diferentes valores calculados en los procesos previos.
6. El panel de información muestra datos a cerca de cada proceso que se realice.
7. El botón “Gráficas” muestra en una nueva ventana todas las gráficas obtenidas en los procesos anteriores.



8. En el panel envío de datos se presentan los botones “Agregar ruido” y “Enviar datos”.
9. En el panel Bits de entrada se muestran los bits que se generaron y se cargaron para iniciar los procesos.
10. En el panel proceso, se muestra el nombre del proceso, el número y los bits que se han generado después de cada proceso.
11. En el panel gráficos se tienen dos pantallas, en la primera muestra las secuencias de los bits que se van generando después de cada proceso. En la segunda se muestra la celda en la que se encuentra el UE, en el momento de realizar el enlace, ya sea ascendente o descendente, en el primer caso este será aleatorio, en el segundo caso dependerá de los valores de PLI y PLCIG que se ingresen, y se procesen en el *Scrambling*.

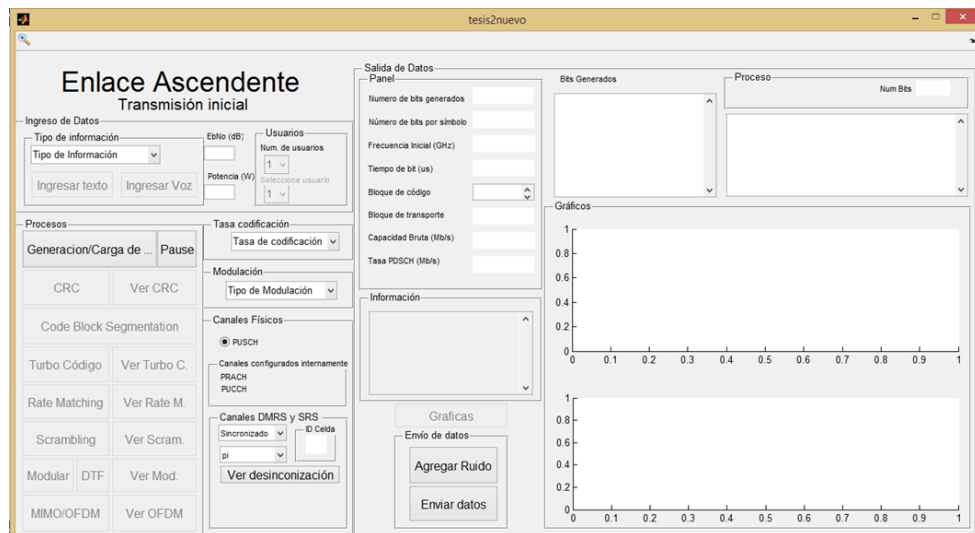
A.2.2. Ventana de Recepción



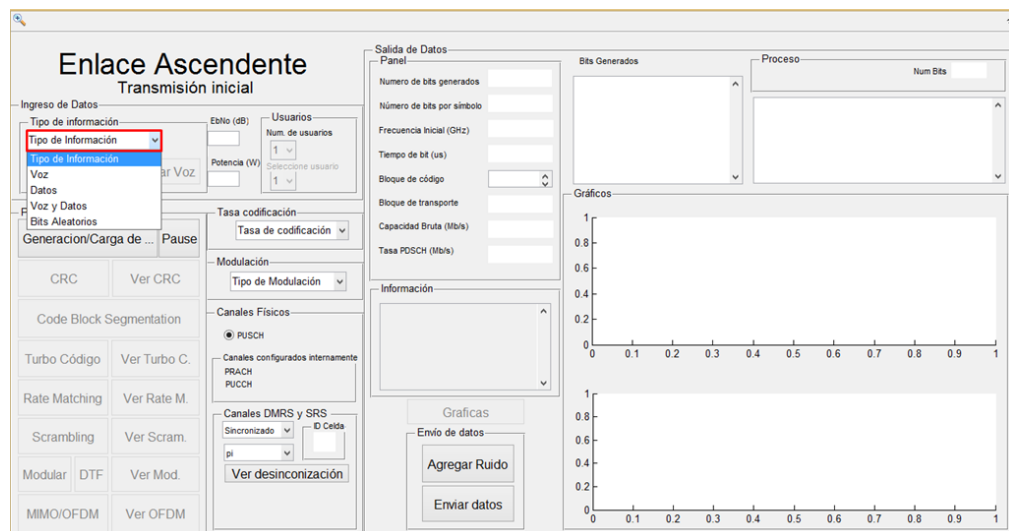
1. El panel de datos recibidos muestra el número de bits recibidos y de cada proceso que se realice, así como los bits que se obtienen, adicional muestra los datos CQI y MSC y los canales que han sido procesados.
2. El panel de gráficos contiene dos pantallas, en la primera se muestra al inicio las subportadoras recibidas, y posteriormente el tren de bits que se obtiene después de cada proceso, en la segunda pantalla se muestra en BER que se obtiene después de cada proceso, con su respectiva modulación.
3. En el panel de información se muestran datos a cerca de cada proceso, y al final de los mismos, se indica si se ha enviado un ACK o un NACK.
4. En el panel del título se indica el tipo de enlace que se está simulando, pudiendo ser ascendente o descendente
5. En el panel de procesos, se encuentran cada uno de los procesos que se pueden ejecutar en la recepción de los datos.
6. En el panel exportación de datos, se tiene las opciones para guardar en un archivo de Excel los datos generados en la simulación realizada, también se puede abrir un archivo generado anteriormente.
7. En el panel mensaje recibido se observa el texto que se ha enviado en el caso de haber escogido este tipo de información para la simulación.
8. Finalmente, se tiene el botón “Nueva transmisión” que sirve precisamente para realizar una nueva simulación.

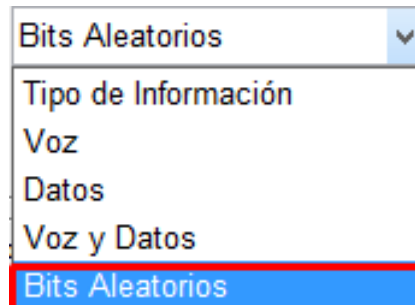
A.3. Enlace Ascendente

1. Al seleccionar la opción “Enlace ascendente”, se abre una ventana con un panel de opciones, tanto para datos de entrada como para datos de salida.

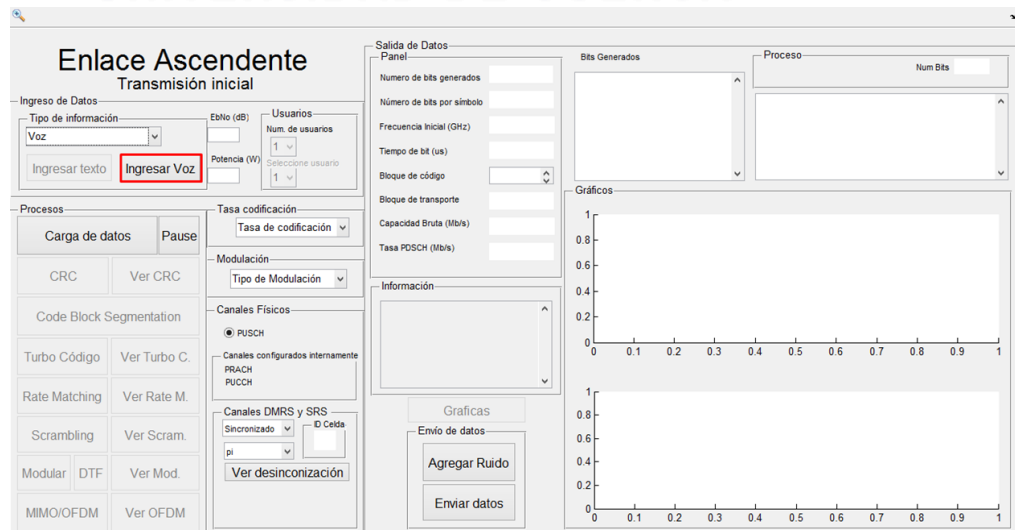
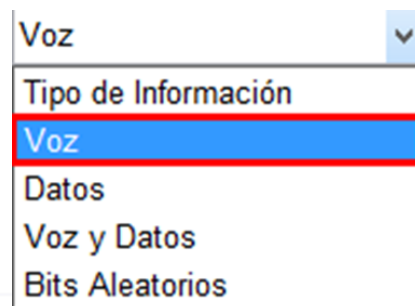


2. El primer parámetro que se configura es el tipo de información a ingresar, para lo cual se despliega la lista de las opciones “Tipo de información”, en la que se muestra los tipos de información que se pueden ingresar.

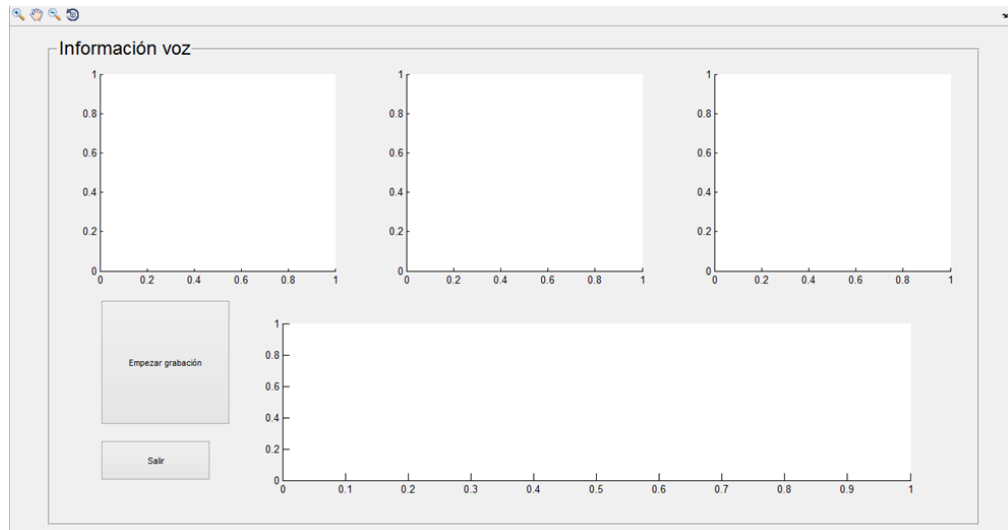




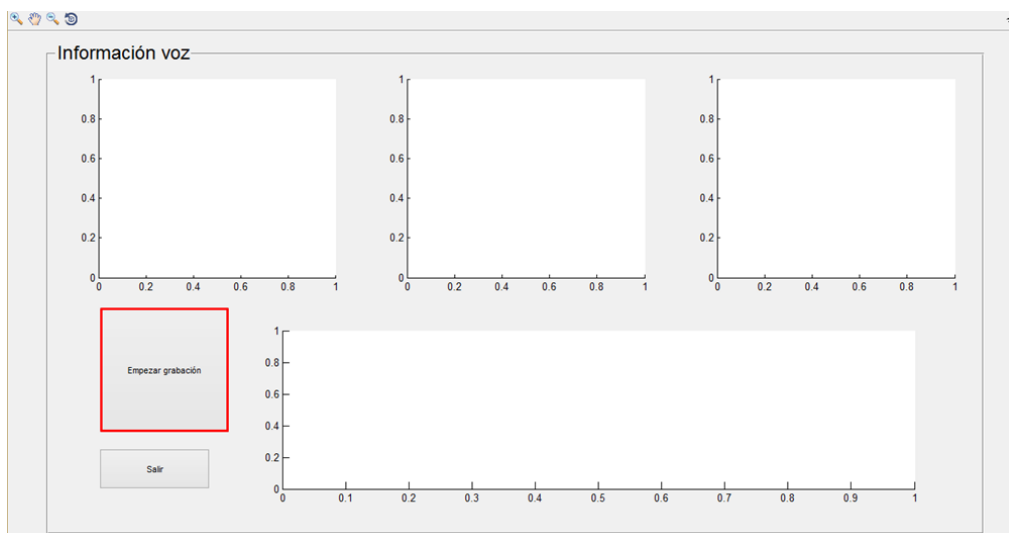
Si se escoge la opción “Voz”, se desbloquea el botón “Ingresar voz”.

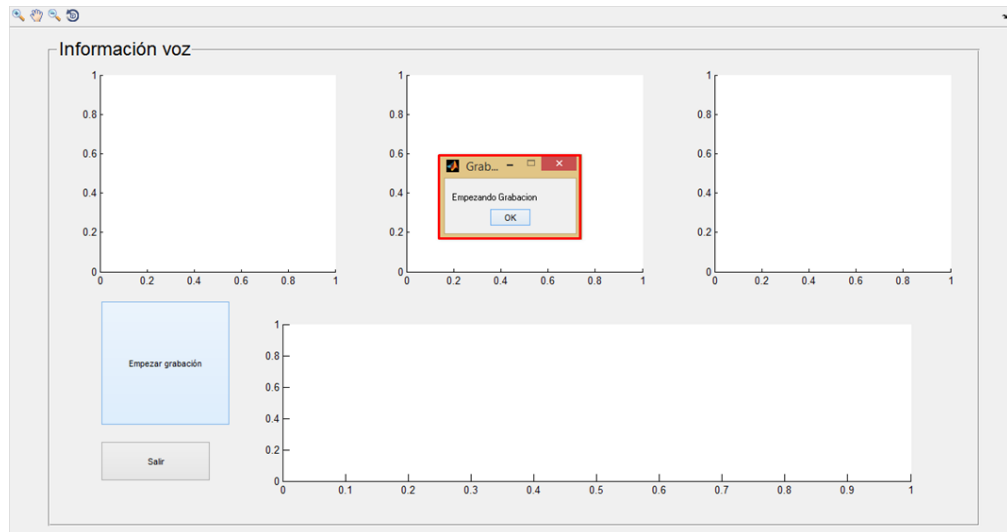


Al seleccionar el botón “Ingresar voz”, se abre la siguiente ventana, en la cual podremos grabar un mensaje de voz de 1 segundo de duración, el cual se codifica y se procesa en los siguientes pasos.

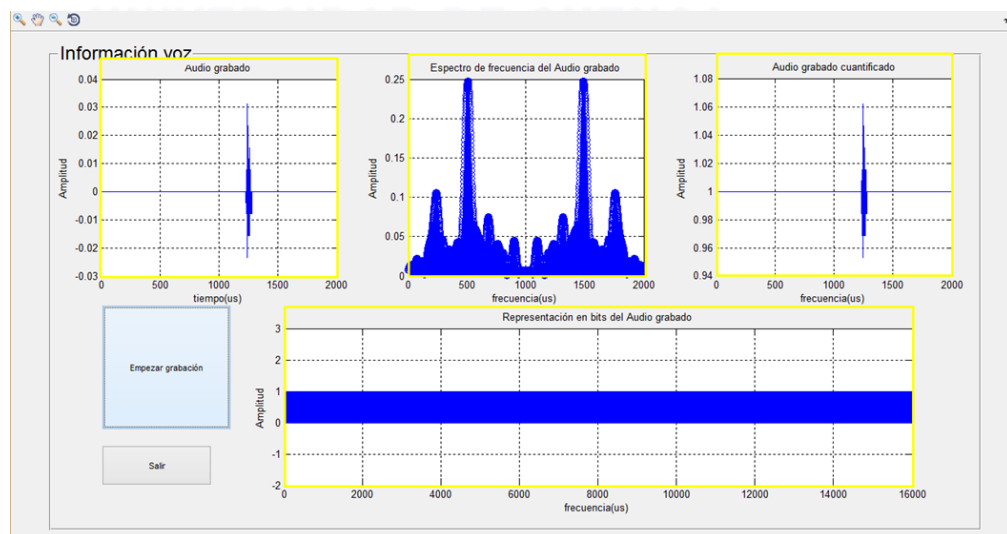


Para empezar la grabación se pulsa el botón “Empezar grabación”, y muestra el mensaje “Empezando grabación” para empezar a hablar y grabar el mensaje.



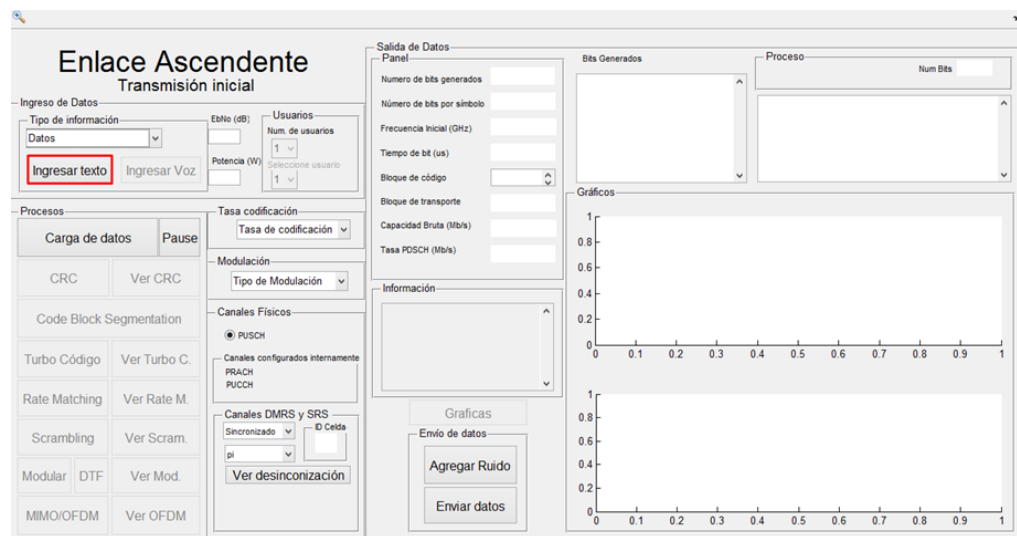
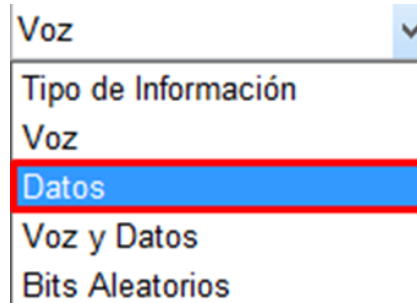


Una vez grabado el audio, se muestran los gráficos del audio adquirido, el espectro de frecuencia, el audio cuantificado, y finalmente, la representación en bits del audio.

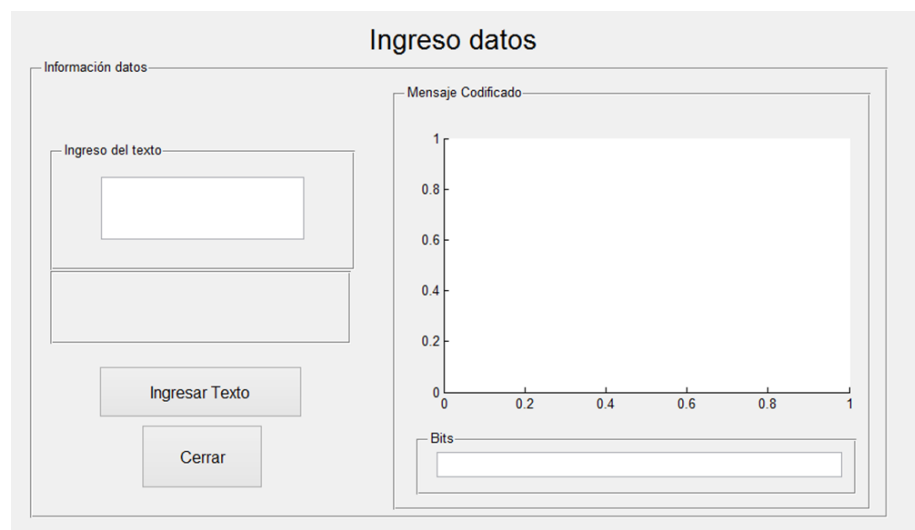


Finalmente, se pulsa el botón “Salir”, para cerrar la ventana.

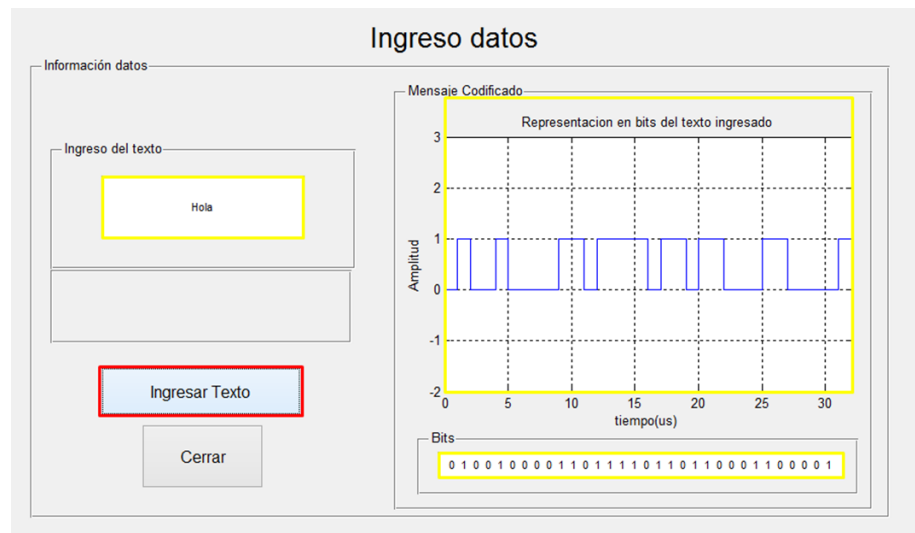
Si se escoge la opción “Datos”, se desbloquea el botón “Ingresar datos”.



Al seleccionar el botón “Ingresar datos”, se abre la siguiente ventana, en la cual se ingresa un texto, de cualquier longitud, el mismo que es codificado y procesado posteriormente.

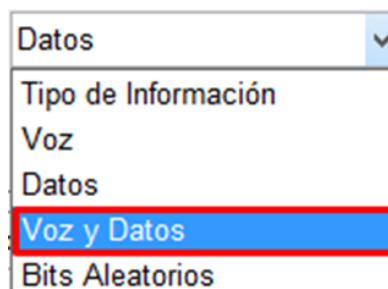


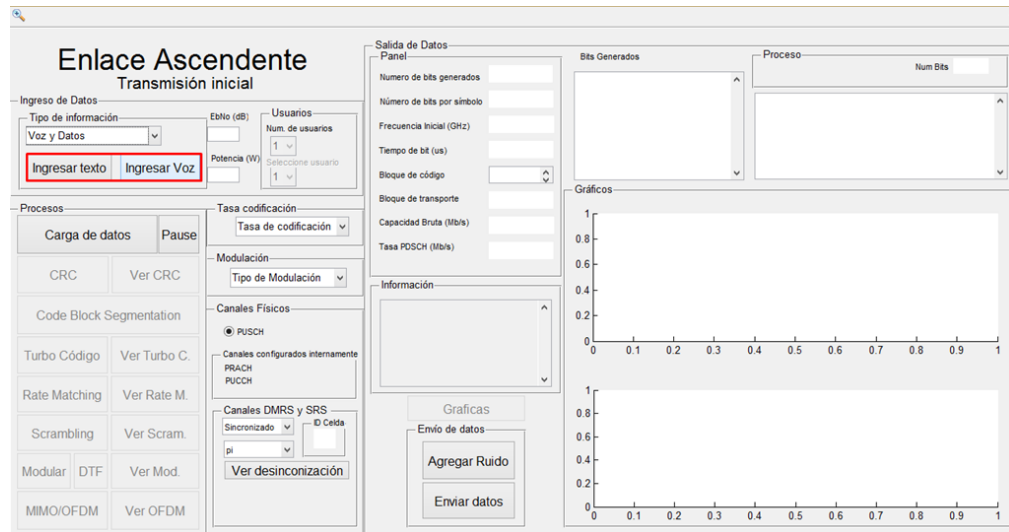
Primero se ingresa el texto y posteriormente se pulsa el botón “Ingresar texto”, y se muestra la codificación del mensaje.



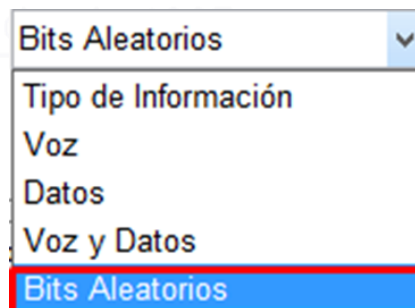
Finalmente, se pulsa el botón “Cerrar”, para cerrar la ventana.

Si se elige la opción “Voz y datos”, se nos desbloquean los botones, “Ingresar voz” e “Ingresar datos”, y se repiten los pasos indicados anteriormente.

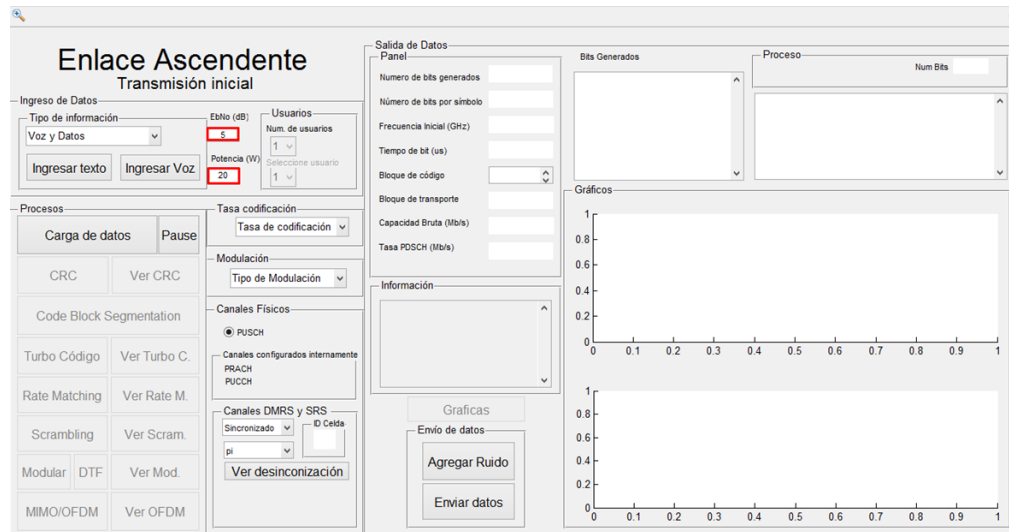




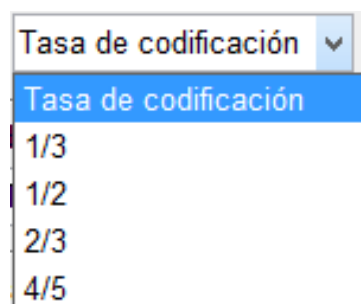
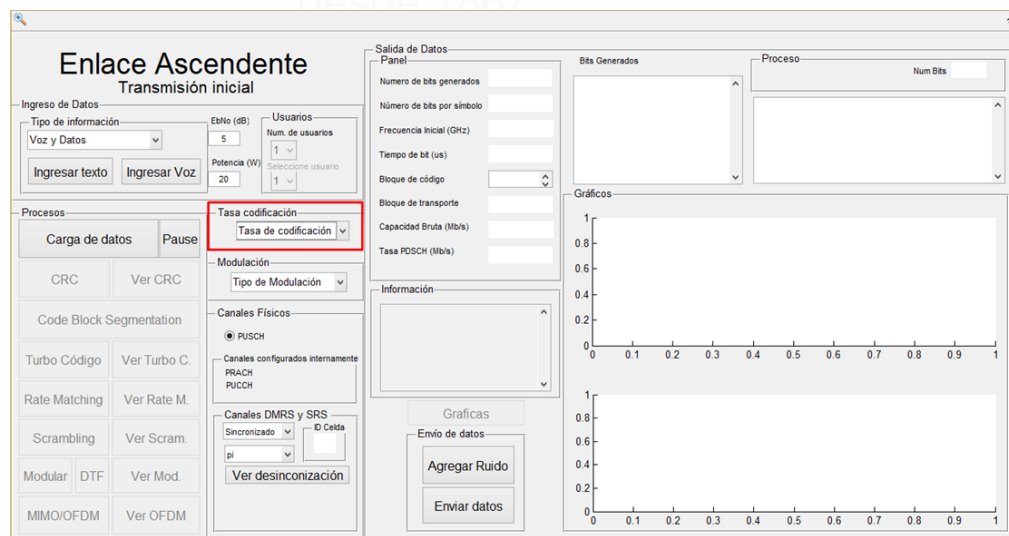
Si se elige la opción “Bits aleatorios”, se cargan bits de forma aleatoria cuando se presione el botón “Generación de bits”, esto solo se podrá hacer una vez configurado las otras opciones que se explican en los siguientes puntos, los bits se generan hasta que se presione el botón “Pausa”.



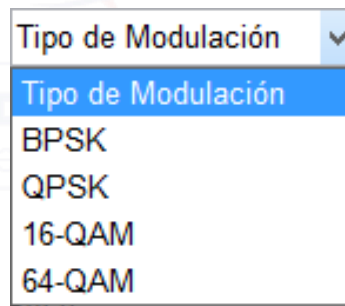
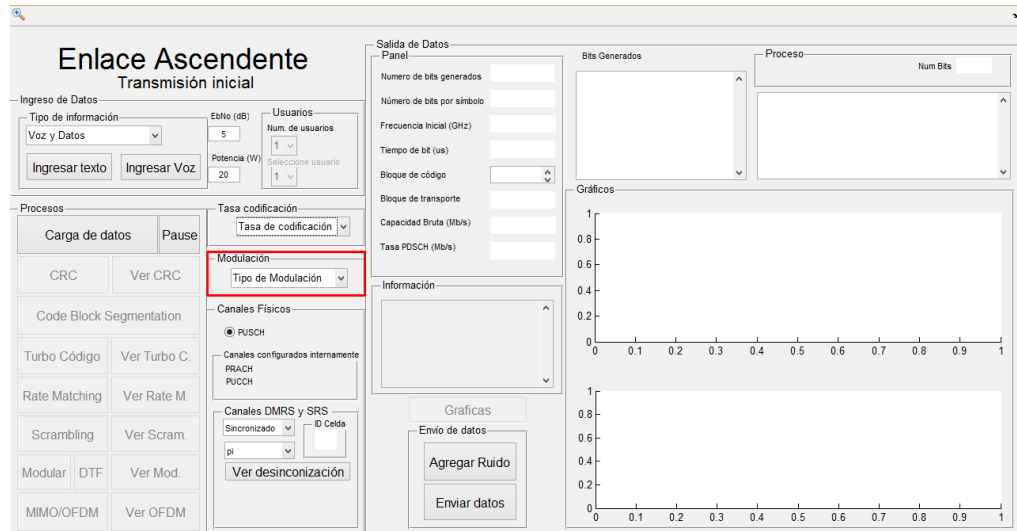
3. Una vez seleccionado el tipo de información, se ingresan los valores de E_bN_0 y de potencia de la señal, deseados para la simulación.



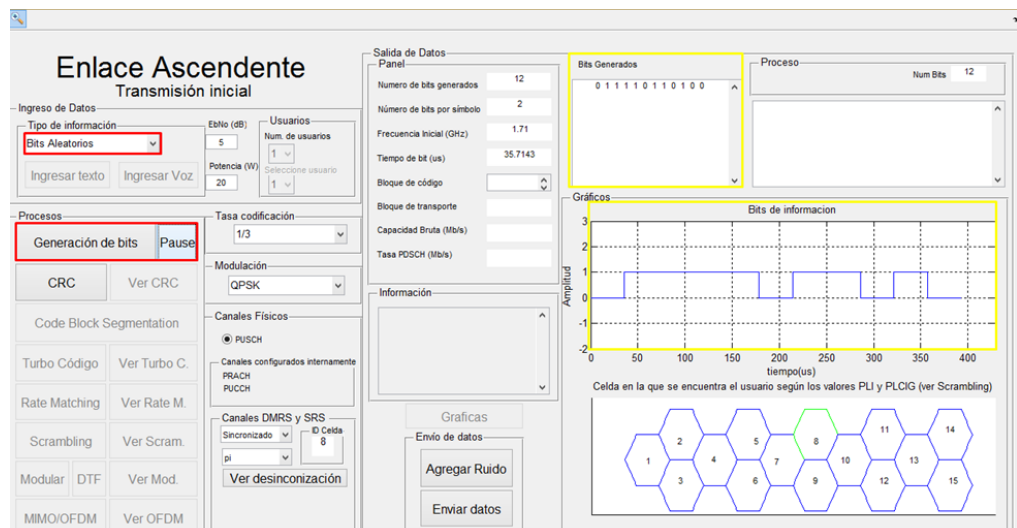
4. A continuación se elige la tasa de adaptación, para esto, se despliega la lista de opciones “Tasa de codificación”.

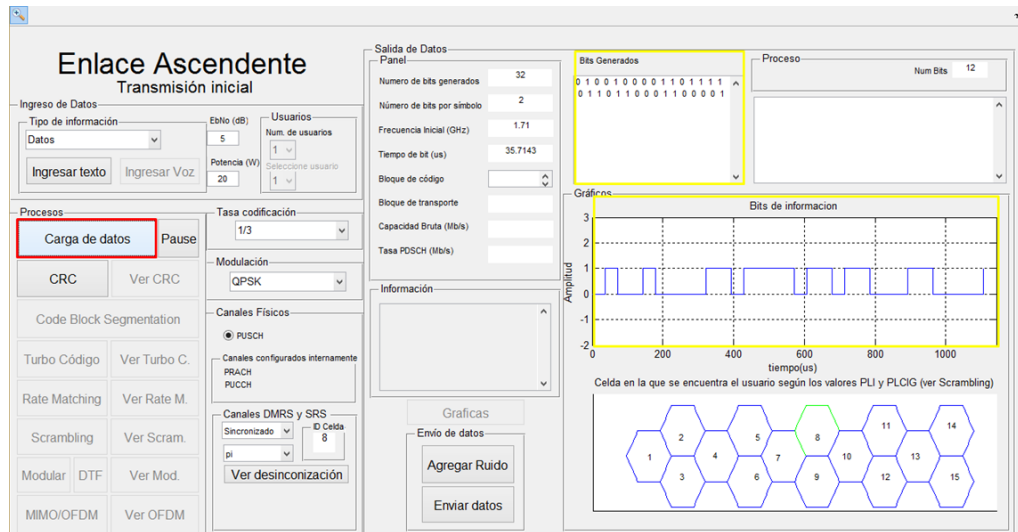


5. Posteriormente se elige el tipo de modulación a utilizar en la simulación, para esto se despliega la lista de opciones “Tipo de modulación”.



6. Generados los pasos anteriores se pulsa el botón “Cargar Datos” o “Generar datos” si se escoge la opción de bits aleatorios. Al presionar este botón se inician los procesos, y se obtienen los primeros datos de salida.

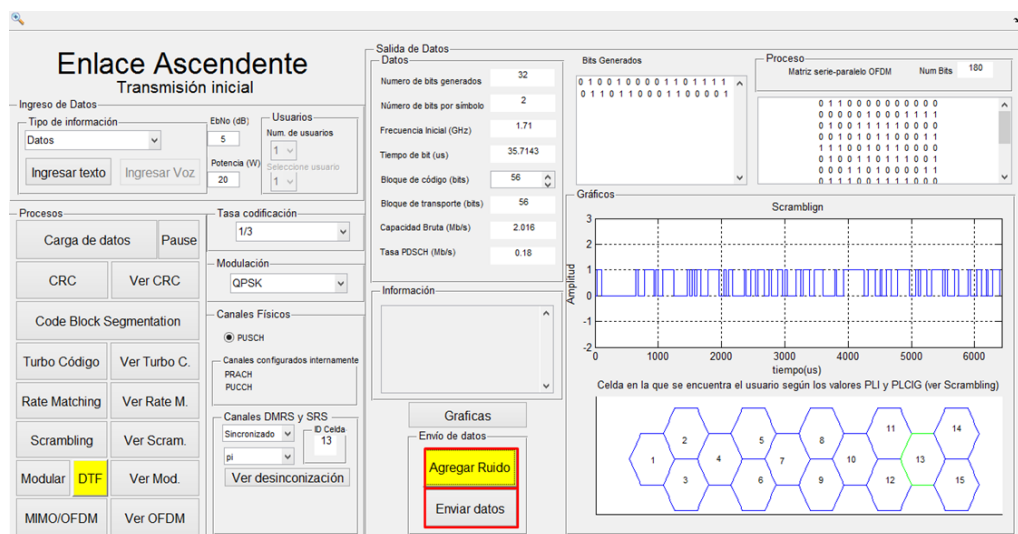




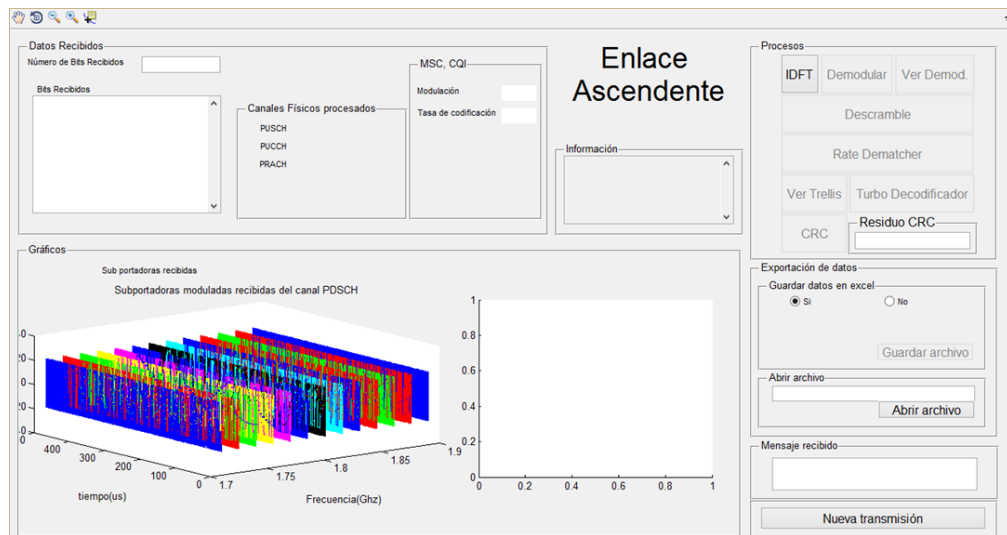
En este caso, se selecciona la opción “Datos” en “Tipo de información, para continuar con el manual.

Se observa que se desbloquea el botón “CRC”, al pulsar el mismo se muestra el resultado de esta operación, y se desbloquea el siguiente botón, y así sucederá sucesivamente con los siguientes botones del panel “Procesos”.

Una vez que se llega al proceso “MIMO/OFDM”, seleccionamos el botón “Agregar Ruido”, si se quiere simular un canal con ruido AWGN. Finalmente, se pulsa el botón “Enviar datos”.

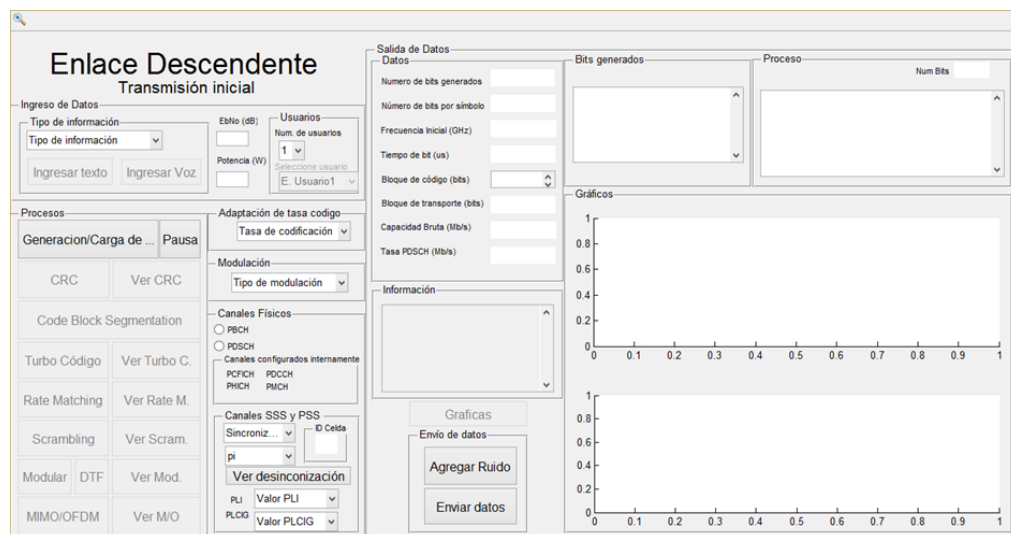


- Una vez enviados los datos, se abre la siguiente ventana, en donde se realizan los procesos paso a paso como en la ventana anterior, en el caso de enlace ascendente no se debe configurar ningún parámetro para la recepción y está disponible el botón “IDTF”.

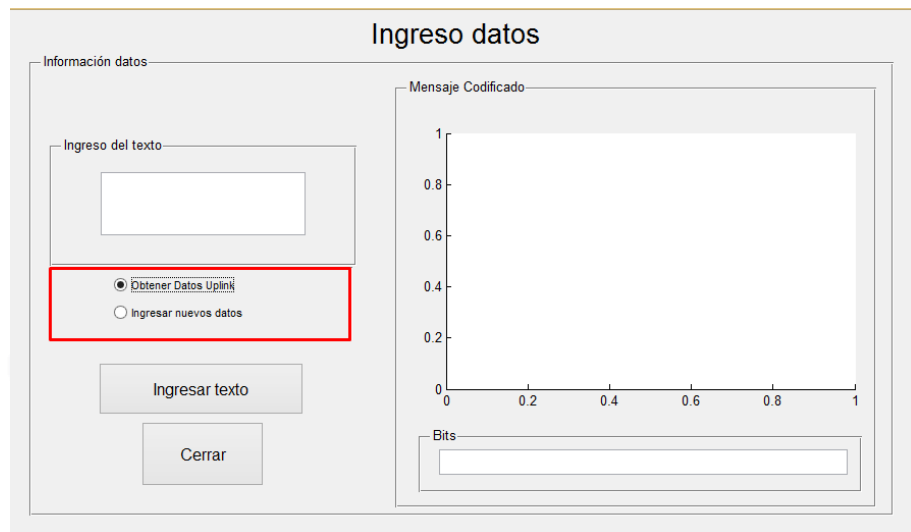


A.4. Enlace Descendente

- Al elegir la opción “Enlace descendente” se abre la siguiente ventana, en donde se tiene un panel de opciones, tanto para datos de entrada como para datos de salida.

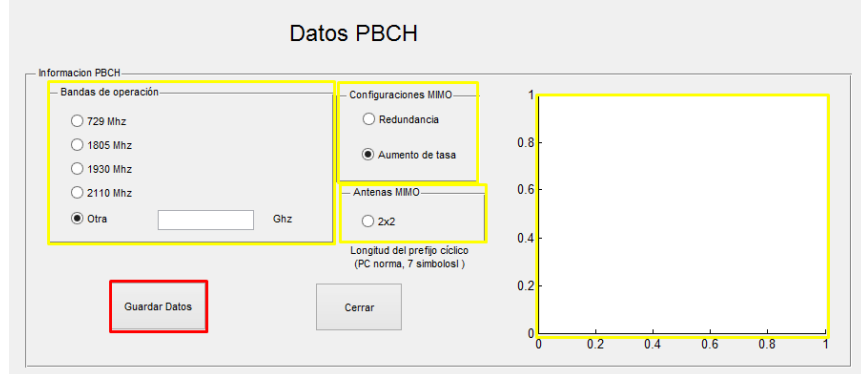


2. El primer parámetro que se configura es el tipo de información a ingresar, para lo cual se despliega la lista de opciones “Tipo de información”, al igual que se hizo en el enlace ascendente. (Ver 2. Enlace ascendente).
- Existen puntos que cambian con respecto al enlace ascendente, uno de ellos es al ingresar texto, en donde se tiene la opción de adquirir el texto que fue enviado previamente en el enlace ascendente, caso contrario, se puede ingresar un nuevo texto como se explicó anteriormente.

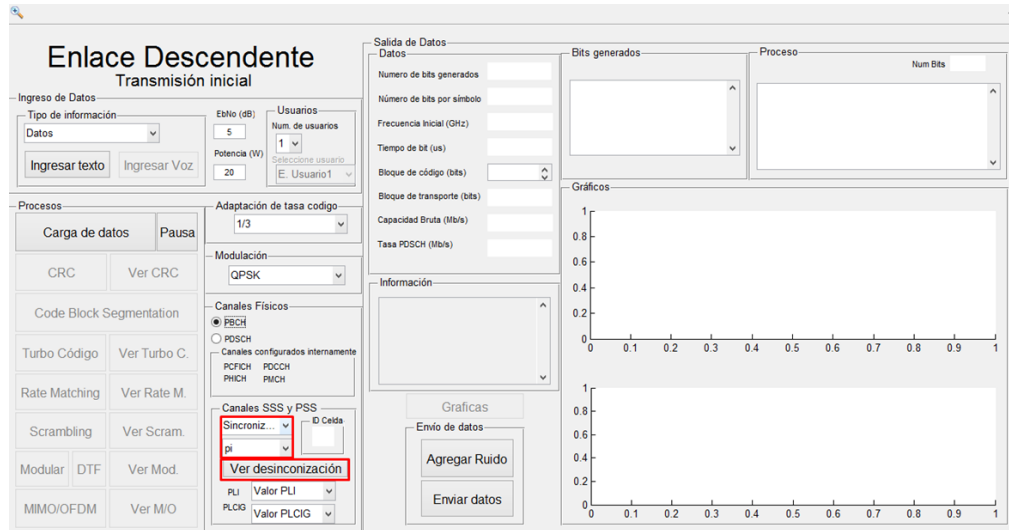


- El segundo punto, es el de número de usuarios, en este tipo de enlace se puede simular, hasta 2 usuarios que procesa la antena, mediante, primero se selecciona el número de usuarios y posteriormente el usuario con el cual se cargan los datos; se debe cargar y configurar datos para cada usuario antes de realizar el proceso “MIMO/OFDM”.
- Otro de los puntos que se configuran adicionalmente, son los valores PLI y PLCIG, las señales de sincronización y el canal PBCH.

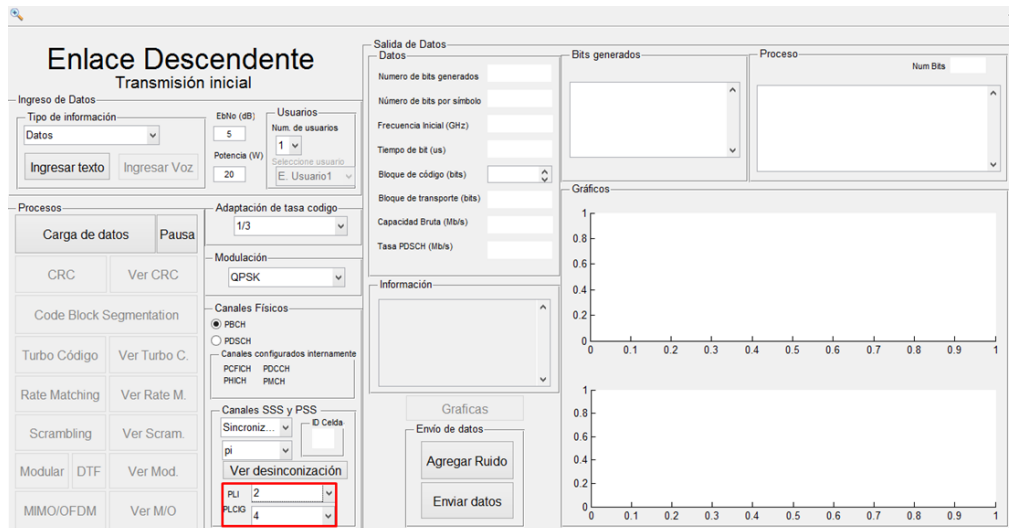
- Para configurar el canal PBCH se marca la opción de este canal y se abre la siguiente ventana:



- Aquí se configura la banda base de operación en la que se desea realizar la simulación; se puede seleccionar cualquiera de los valores disponibles o ingresar otra banda. Para el caso de enlace ascendente la banda de frecuencia se asigna de acuerdo a la escogida en enlace descendente, de no haber simulado previamente el enlace descendente, la banda está establecida en 1700 MHz.
- También se elige la configuración MIMO, redundante o aumento de tasa.
- Finalmente, se selecciona el número de antenas que se desea, y se pulsa el botón “Guardar datos” y luego el botón “Cerrar”, para cerrar la ventana.
- En cuanto a las señales de sincronización, el programa permite simular una desincronización en la transmisión, para esto, se selecciona la opción desincronizar y posteriormente se selecciona el ángulo de desfase que se quiere aplicar.

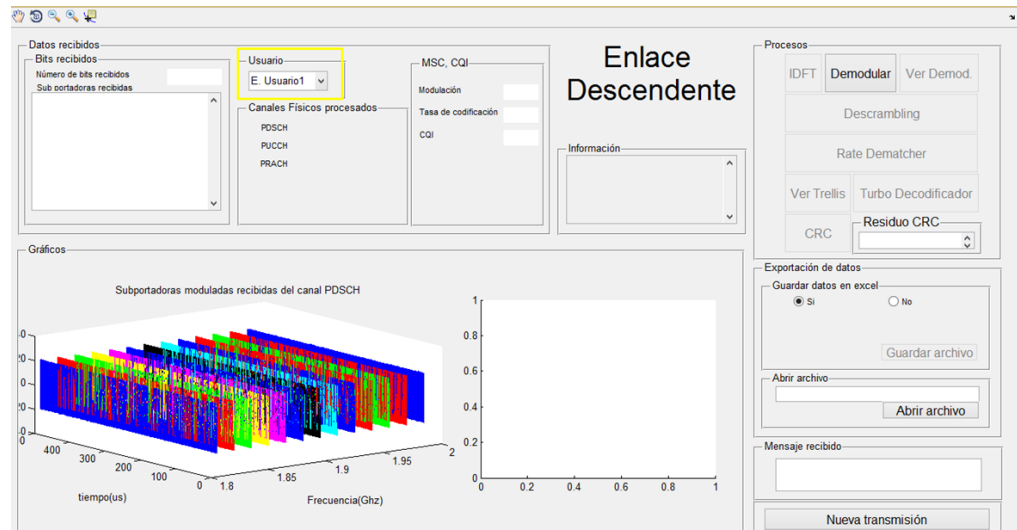


- Esta opción también se la puede realizar en enlace ascendente.
- Para poder ver la desincronización se pulsa el botón “Ver desincronización”.
- Finalmente, se configuran los valores PLI y PLCIG, para determinar la identidad de la celda en la que se ubica.



3. En el siguiente paso, manteniendo seleccionada la opción “PBCH”, se procede a cargar los datos y realizar todos los procesos hasta la modulación.
4. Posteriormente se selecciona la opción PDSCH y se realizan nuevamente todos los procesos al igual que en el enlace ascendente.

5. En la recepción, la diferencia con el enlace ascendente es que, en el caso que se seleccione dos usuarios, se debe seleccionar un usuario a la vez para procesar los datos, primero usuario 1 y luego usuario 2 o viceversa.

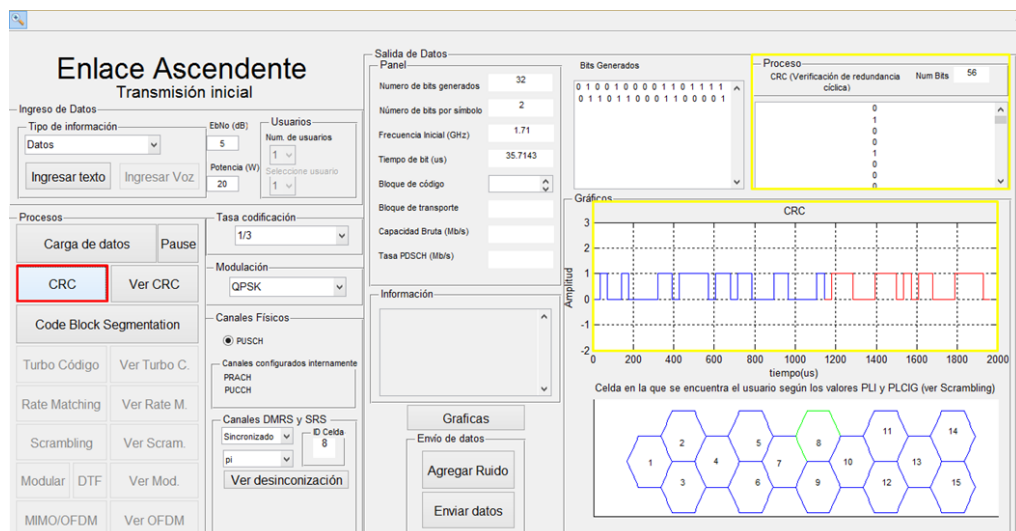


A.5. Procesos

A.5.1. Procesos de envío

Codificación CRC

El primer proceso después de haber cargado o generado los datos, es la codificación “CRC”, se pulsa el botón para ejecutarlo. Se visualiza unos bits de color rojo, estos bits representan los 24 bits de paridad agregados por el CRC.



Para ver este proceso en detalle, se pulsa el botón “Ver CRC” y se abre una pantalla como la siguiente, en donde se pulsa el botón “Iniciar” y posteriormente el botón “Siguiente paso” hasta terminar el proceso.



Verificación por redundancia cíclica

Datos de entrada

Mensaje

Polinomio generador

$g(x) = x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$ L=24

Operaciones

Polinomio Inicial

Resultado iteración anterior

Generador

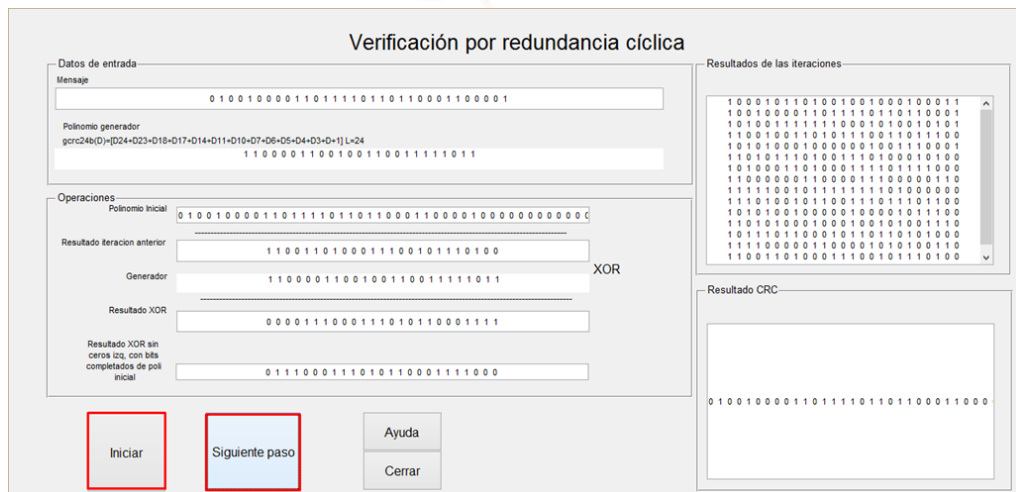
Resultado XOR

Resultado XOR sin
ceros izq. con bits
completados de poli
inicial

Resultados de las iteraciones

Resultado CRC

Iniciar Siguiente paso Ayuda Cerrar



Verificación por redundancia cíclica

Datos de entrada

Mensaje

Polinomio generador

$g(x) = x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$ L=24

Operaciones

Polinomio Inicial

Resultado iteración anterior

Generador

Resultado XOR

Resultado XOR sin
ceros izq. con bits
completados de poli
inicial

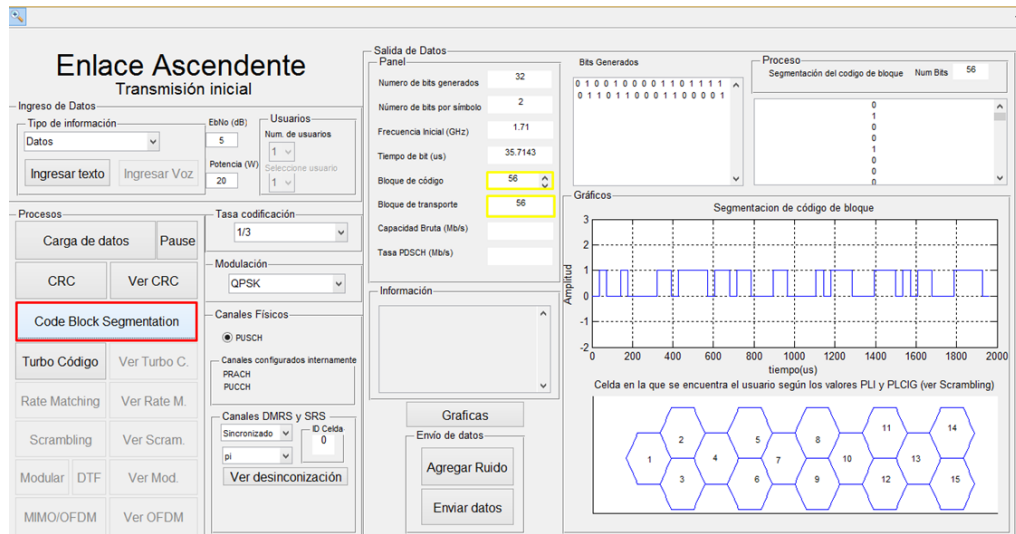
Resultados de las iteraciones

Resultado CRC

Iniciar Siguiente paso Ayuda Cerrar

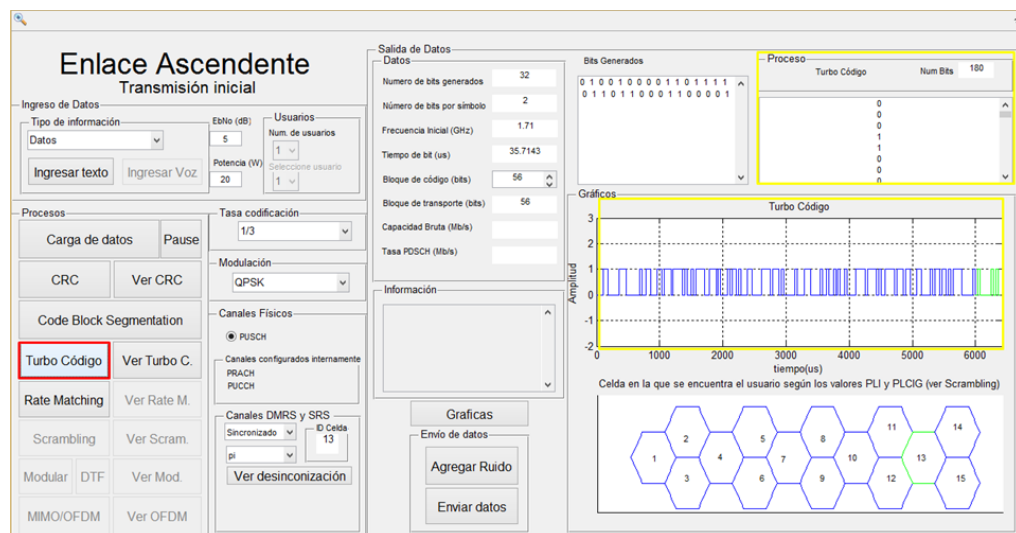
Segmentación de código de bloque

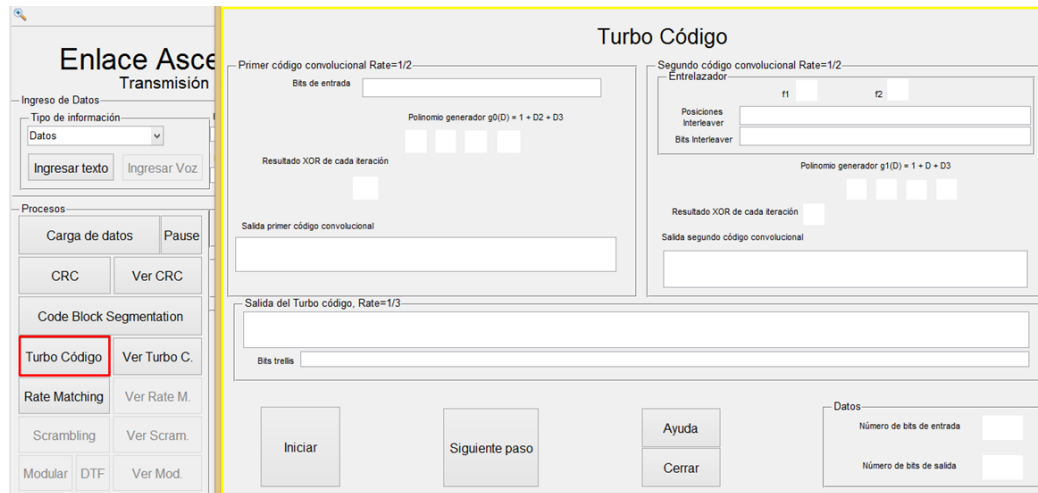
El segundo proceso es la “Segmentación de código de bloque”, los resultados de este proceso se visualizan en las casilla de “Bloque de código”, y “Bloque de recursos”. Como se muestra a continuación:



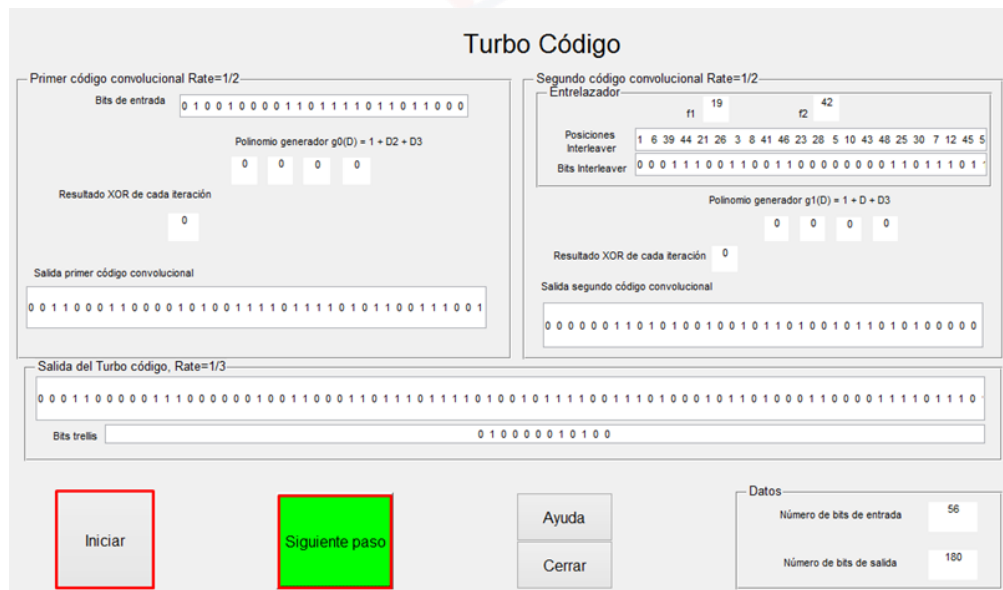
Turbo código y Tail biting

El tercer proceso que se realiza es el “Turbo código”, o el “*Tail biting*” para el caso del canal PBCH en enlace descendente, para esto se pulsa el botón “Turbo código”. Se muestran unos bits de color verde, estos bits representan los bits Trellis a la salida de turbo codificador. Para visualizar el proceso paso a paso se pulsa el botón “Ver Turbo C”.





Dentro esta ventana de igual manera, se presiona el botón “Iniciar”, y con el botón “Siguiete paso”, se observa el proceso.



En cuanto al *Tail biting*, es una opción que se presentará al procesar el canal PBCH en el enlace descendente, se pulsa el botón de este proceso y nos aparecerá la siguiente ventana, en la cual se inicia el proceso con el botón *Iniciar* se continúa el proceso con el botón “Siguiete paso”.

Código convolucional Tail Biting

Proceso

Bits de entrada

Registros ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

dk0

dk1

dk2

Salida Tail Biting

Inicio

Siguiente paso

Ayuda

Cerrar

Datos

Número de bits de entrada

Número de bits de salida

Code Block Segmentation

Tail Biting

Ver Tail B.

Canales Físicos

☒ PBCH

☐ PDSCH

Canales configurados internamente

PCFICH PDCCH

tiempo(us)

0 1000 2000 3000 4000 5000 6000 7000

Código convolucional Tail Biting

Proceso

Bits de entrada

Registros

dk0

dk1

dk2

Salida Tail Biting

Inicio

Siguiente paso

Ayuda

Cerrar

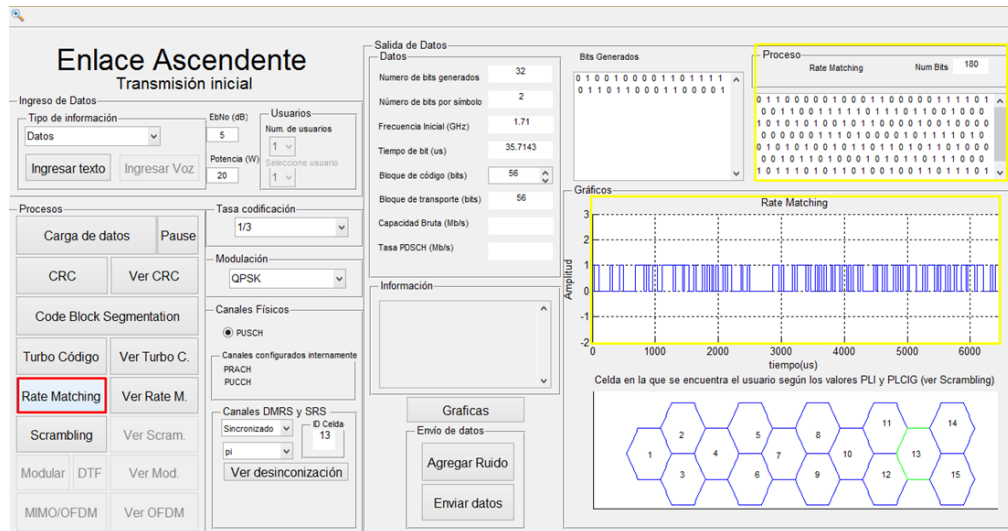
Datos

Número de bits de entrada

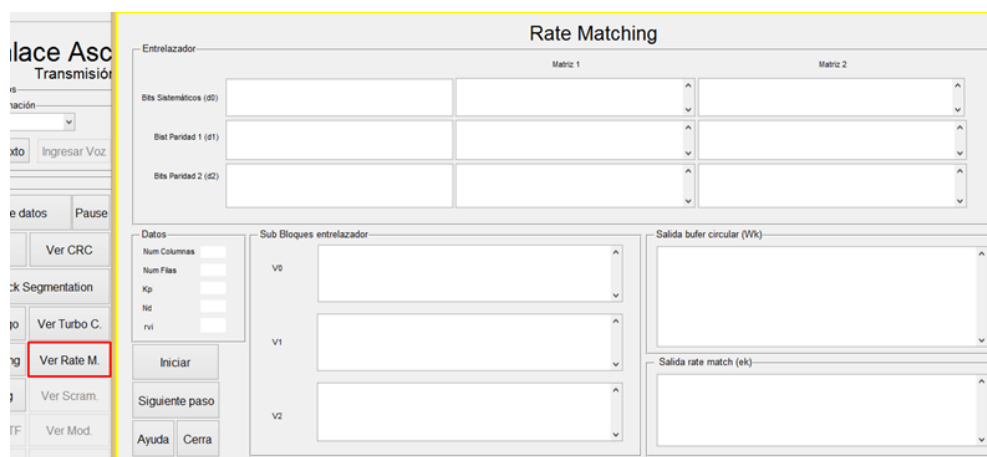
Número de bits de salida

Adaptación de velocidad (*Rate Matching*)

El cuarto proceso es la adaptación de velocidad (*Rate Matching*), para ejecutarlo damos clicse pulsa el botón Rate Matching”.

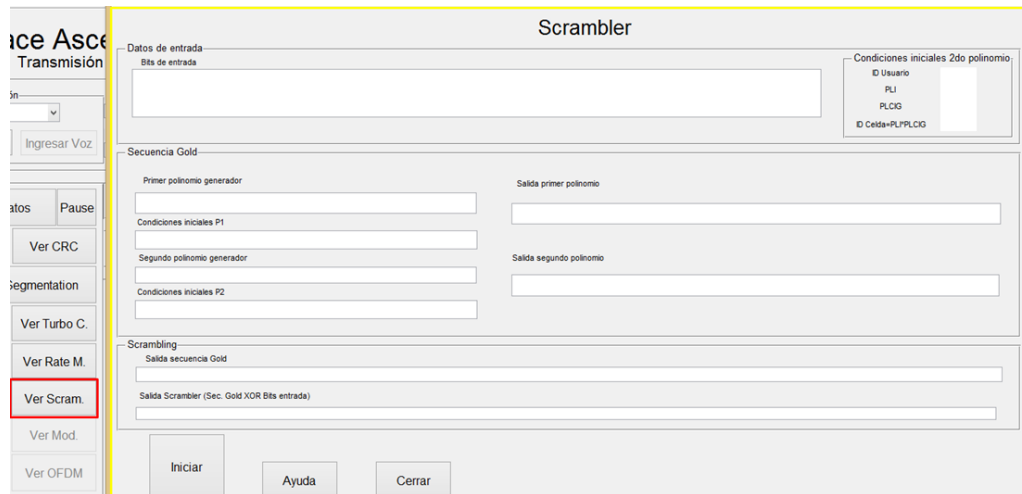


Para ver este proceso paso a paso se pulsa el botón “Ver Rate M.”, se abre la siguiente ventana, de igual manera primero se pulsa el botón “Iniciar” y posteriormente en el botón “Siguiente paso” hasta terminar el proceso. Se pulsa el botón “Cerrar”, para salir.

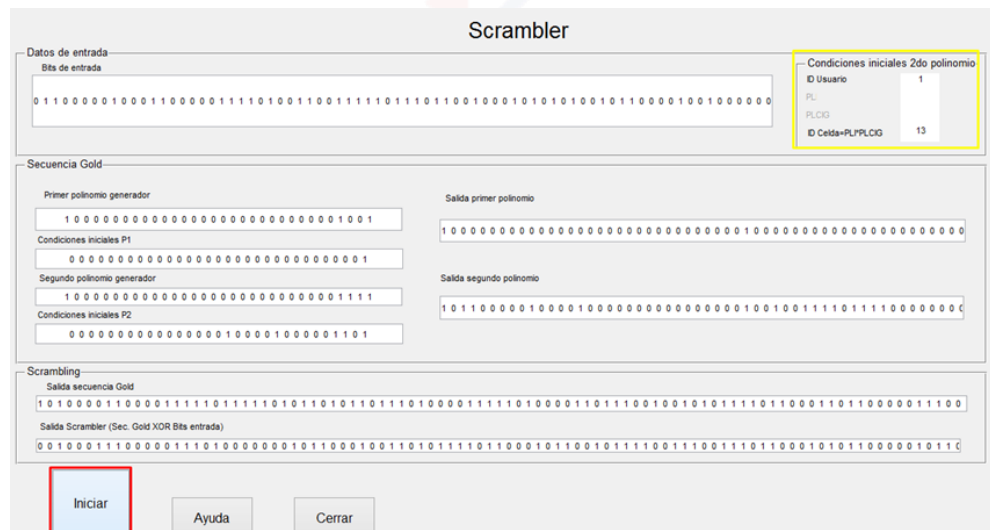




Aleatorización *Scrambling*



Para ejecutar el proceso se pulsa el botón “Iniciar” y se muestra todo el proceso, en este caso no existe la opción “Siguiente paso”.

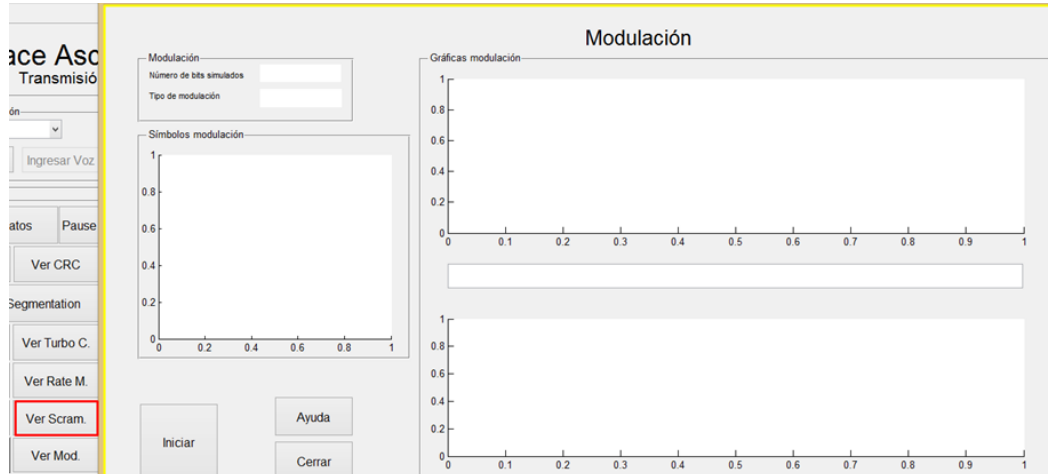
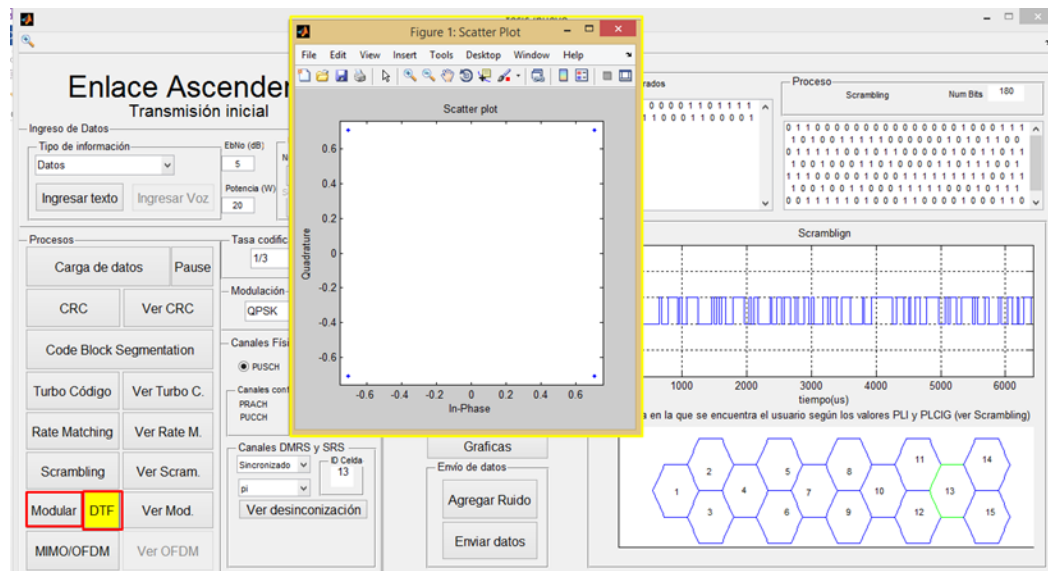


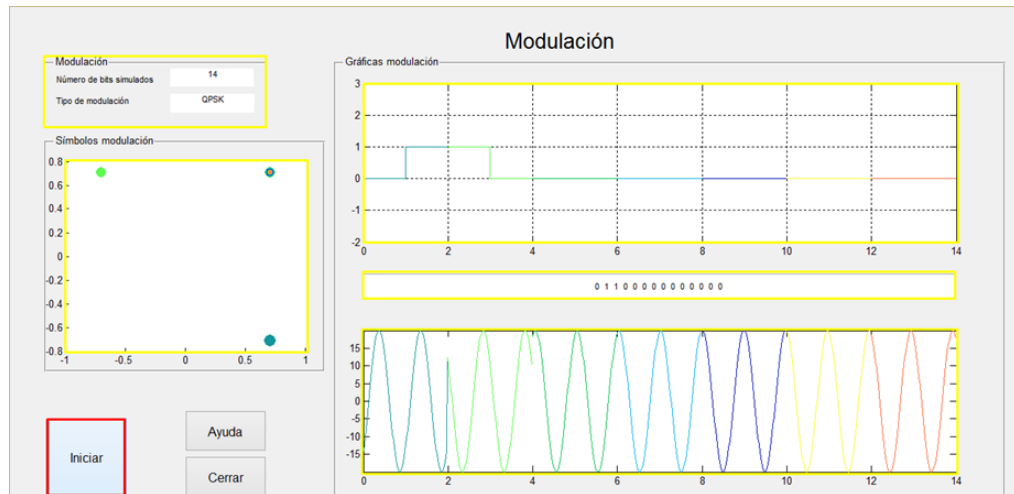
Aquí se puede observar las condiciones iniciales de la secuencia *scrambling* las cuales se muestran en la esquina superior derecha de la ventana.

Modulación

El sexto proceso es la “Modulación”, para esto pulsa el botón con el mismo nombre, inmediatamente se muestra el diagrama de constelación de la modulación escogida. Para ver el proceso de modulación completo, se selecciona el botón “Ver Mod.” y se abre una ventana, en la cual se puede iniciar el proceso, y se puede ver el tren de bits como su modulación, y los símbolos en el diagrama de constelación. Así como el número de bits que se están simulando y el tipo de modulación.

Para el caso de enlace ascendente, posterior a la modulación se debe hacer una precodificación, esto se realiza al presionar el botón “DFT”, el mismo que cambia de color al haber realizado este proceso.

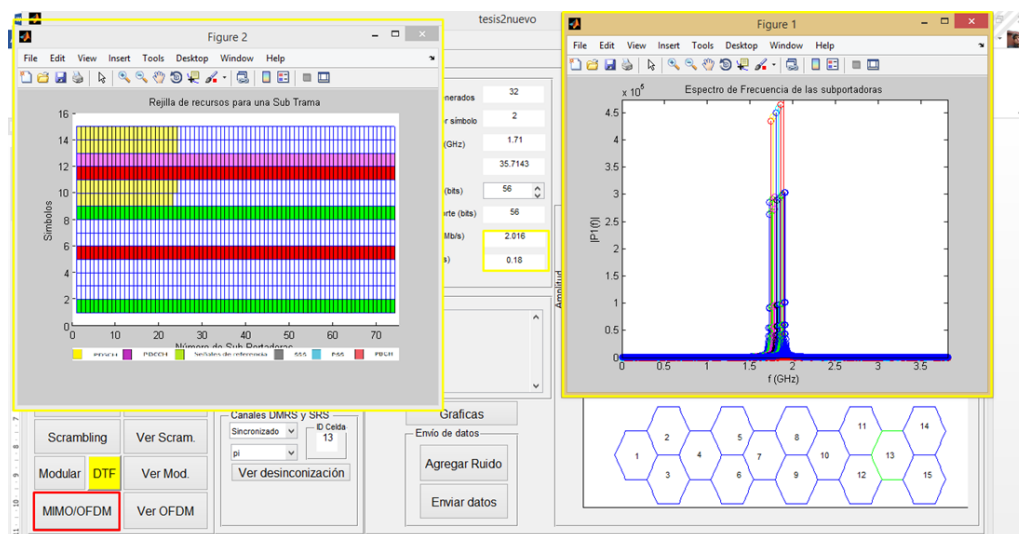




MIMO/OFDM

El séptimo proceso une a dos pasos de LTE, MIMO y OFDM, para ejecutarlos se pulsa el botón “MIMO/OFDM”; mediante el cual se procesa la información y se abren dos imágenes, una con la rejilla OFDM, en donde se visualiza la repartición de recursos, y si se utilizó técnicas MIMO multiantena se ven las diferentes capas de procesamiento. En la segunda imagen se puede ver el espectro de frecuencia de cada una de las subportadoras generadas por OFDM. Además se muestran los cálculos de la capacidad bruta del canal y la tasa de datos PDSCH.

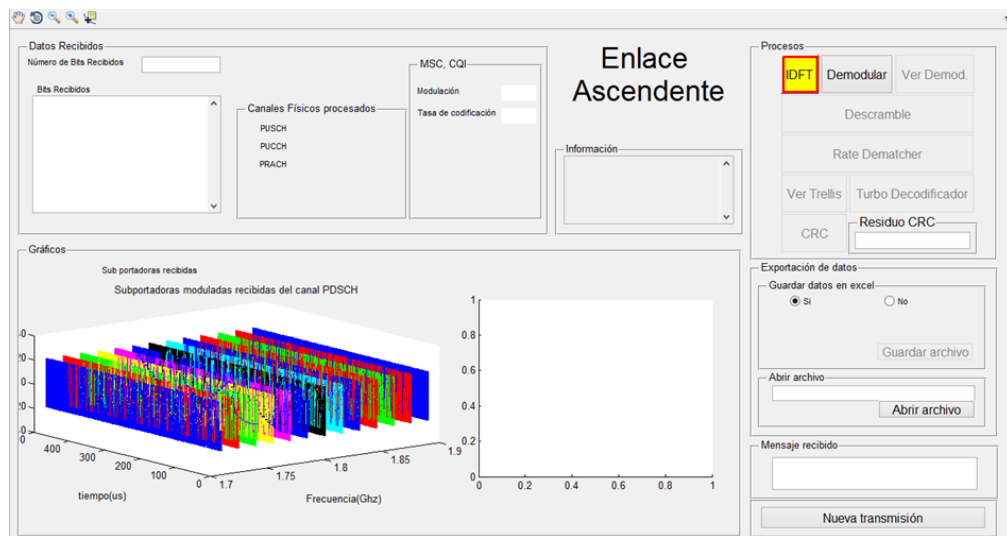
Para observar paso a paso este proceso se pulsa el botón “Ver M/O”. Se nos abrirá la siguiente ventana.



A.5.2. Procesos recepción

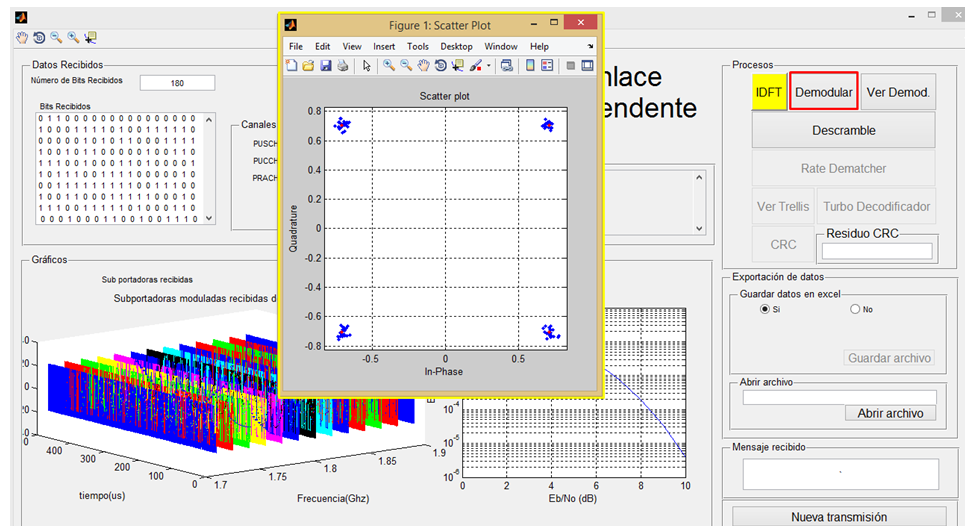
IDFT

El primer proceso en la recepción, en el caso de enlace ascendente, es la IDFT, al seleccionar el botón “IDFT” se ejecuta el proceso y el botón cambia de color, en el caso del enlace descendente este botón no está disponible y el primer proceso es el de demodulación.

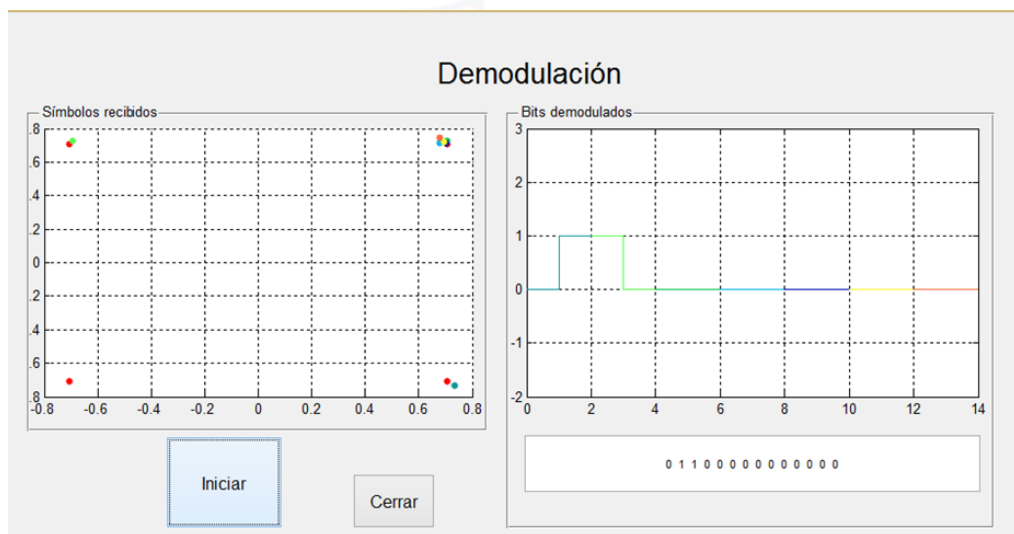
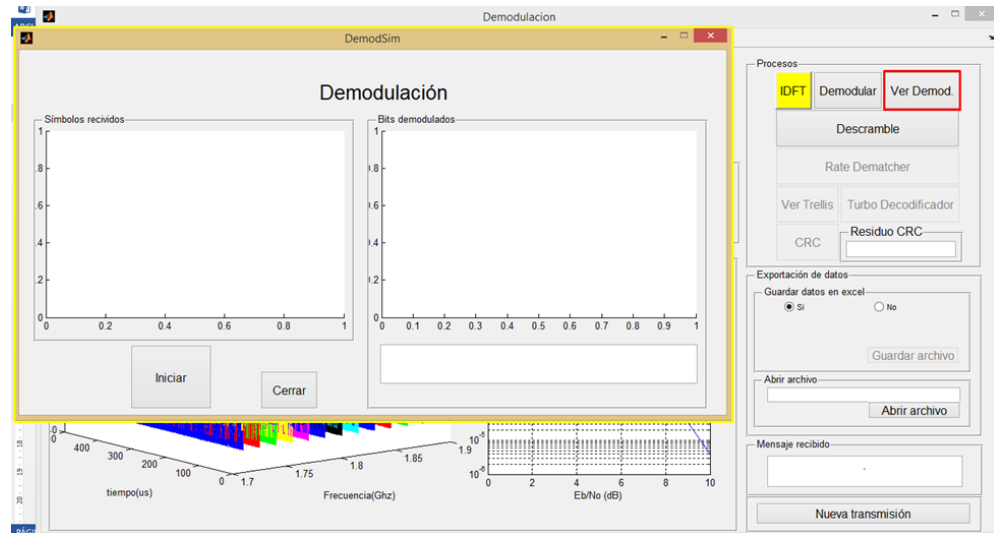


Demodulación

El proceso de demodulación, se lo ejecuta al presionar el botón “Demodular”, y se visualiza el diagrama de constelación con los símbolos afectados por ruido del canal.

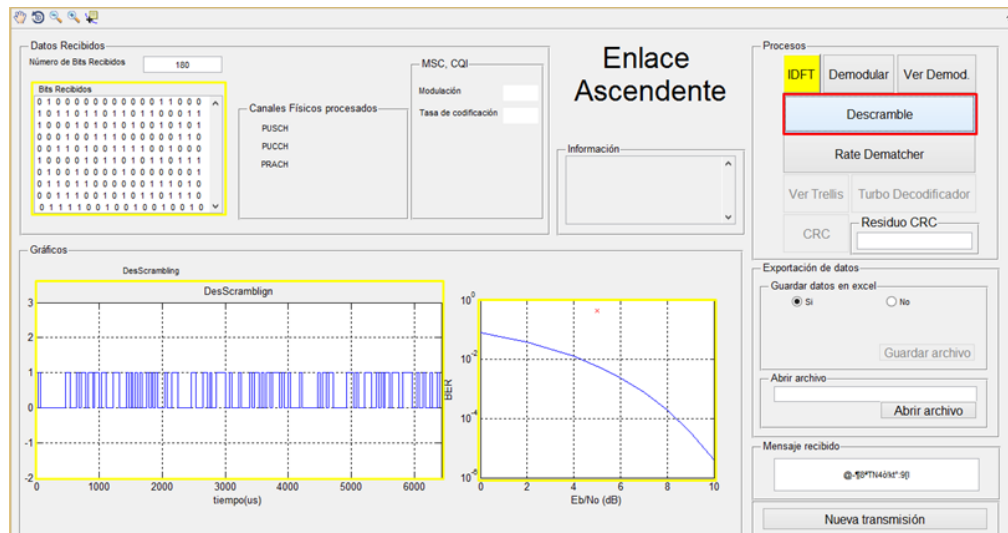


Para ver el proceso de demodulación se presiona el botón “Ver demod.”, y se abre la siguiente ventana, en la cual se ejecuta con el botón “Iniciar”.



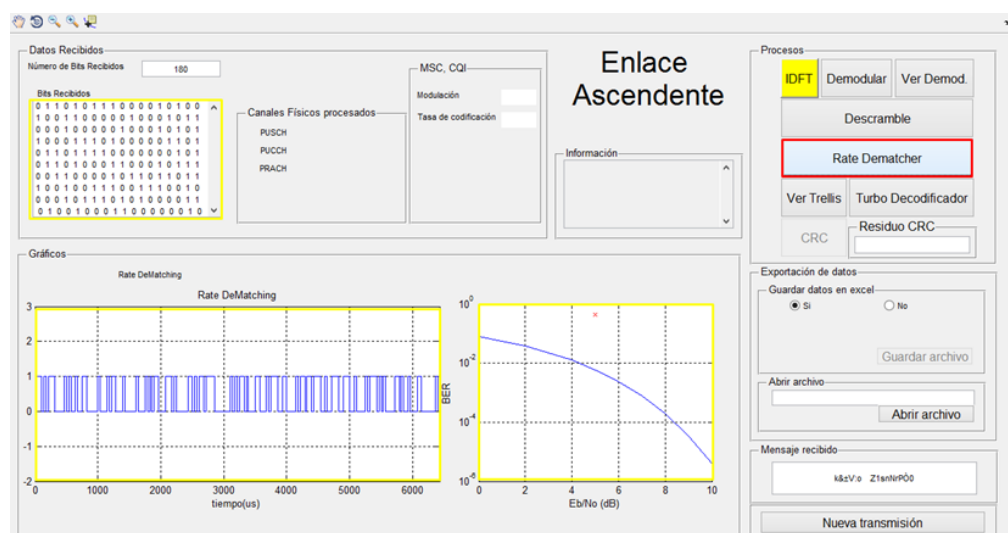
DeScrambling

El tercer proceso es el *desescrambling*, para ejecutarlo se pulsa el botón "Descramble".



Rate dematcher

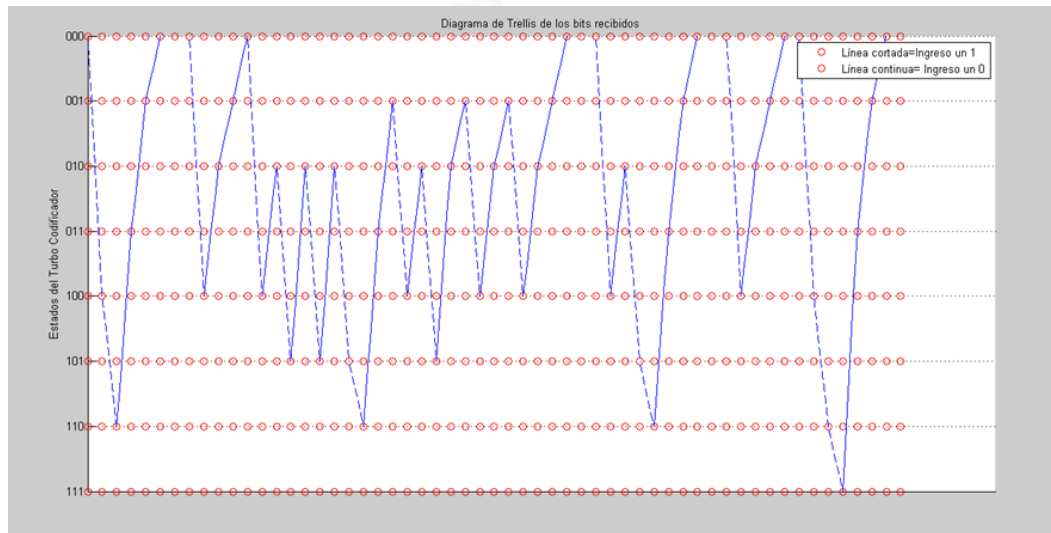
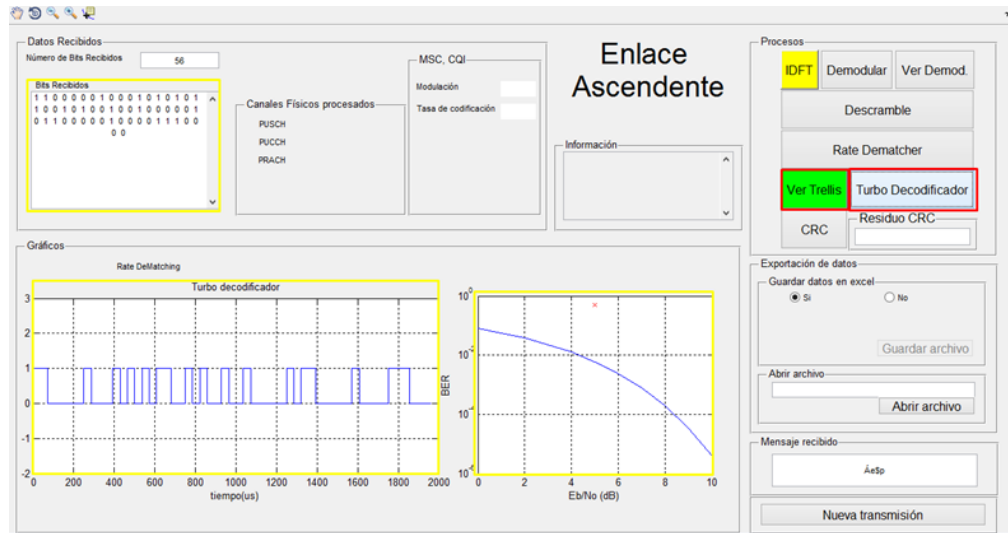
El cuarto proceso es el rate dematcher, para ejecutarlo se pulsa el botón "Rate Dematcher".



Turbo Decodificación.

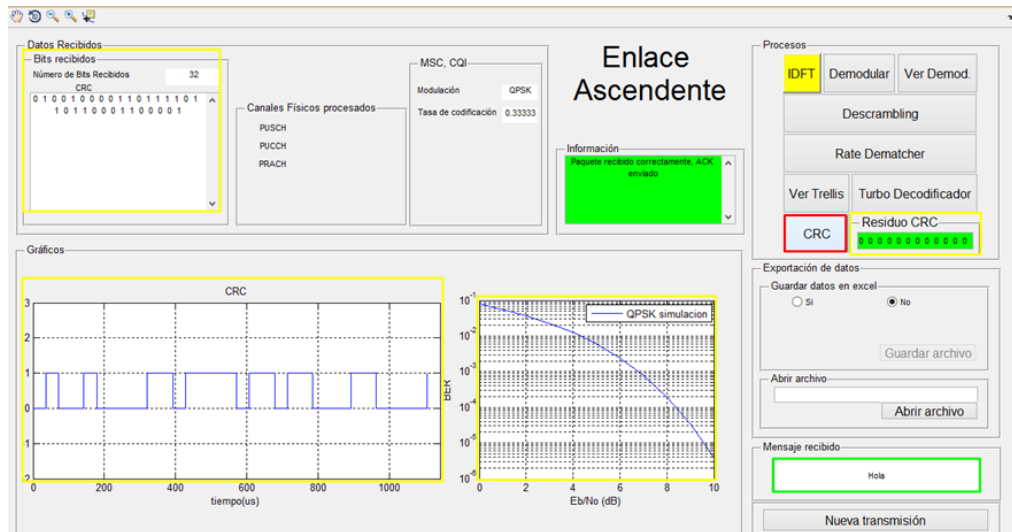
El quinto proceso es la turbo decodificación, en este paso se puede elegir ver el diagrama de Trellis mediante el botón "Ver Trellis", este cambia de color y a

continuación se pulsa en “Turbo Decodificador”, y se abre una ventana con el diagrama de Trellis empezando el proceso.



CRC

El sexto proceso, es el de CRC, para eso se pulsa el botón con el mismo nombre.



Al ejecutar este proceso se puede observar el residuo del CRC, el mismo que junto con el parámetro CQI determinan si se envía un ACK o un NACK, si se envió un texto se lo observa en la pantalla y si es un audio se reproduce.

Apéndice B

Glosario de términos

- **3GPP:** 3rd Generation Partnership Project.
- **AMC:** Adaptive Modulation and Coding.
- **BCH:** Broadcast Channel.
- **BER:** Bit Error Rate.
- **BPSK:** Binary Phase Shift Keying Modulation.
- **CFI:** Control Format Indicator Channel.
- **CQI:** Channel Quality Indicator.
- **CRC:** Cyclic Redundancy Check.
- **DFT:** Discrete Fourier Transform.
- **DLSCH:** Downlink Shared Channel.
- **DMRS:** Demodulation Reference Signal.
- **eNB:** Evolved Node B.
- **EPC:** Evolved Packet Core.
- **E-UTRAN:** Evolved UMTS Terrestrial Radio Access Network.
- **FDD:** Frequency Division Duplex.
- **HSS:** Home Subscriber Server.
- **ISI:** Inter Symbol Interference.
- **LTE:** Long Term Evolution.
- **MAC:** Medium Access Control Layer.
- **MATLAB:** Matrix Laboratory.
- **MCH:** Multicast Channel.
- **MCS:** Modulation and Coding Scheme.
- **MIMO:** Multiple Input Multiple Output.
- **MME:** Mobile Management Entity.
- **OFDM:** Orthogonal Frequency Division Multiple Access.
- **PCCC:** Parallel Concatenated Convolutional Code.



- **PCFICH:** Physical Control Format Indicator Channel.
- **PCH:** Paging Channel.
- **PDCCH:** Physical Downlink Control Channel.
- **PDN:** Packet Data Network.
- **PDSCH:** Physical Downlink Shared Channel.
- **P-GW:** PDN Gateway.
- **PHICH:** Physical Hybrid-ARQ Indicator Channel.
- **PSS:** Primary Synchronization Signal.
- **PUSCH:** Physical Uplink Shared Channel.
- **QAM:** Quadrature Amplitude Modulation.
- **QoS:** Quality of Service.
- **QPSK:** Quadrature Phase-Shift Keying.
- **RNTI:** Radio Network Temporary Identifier.
- **RS:** Reference Signal.
- **SC-FDMA:** Single Carrier Frequency Division Multiple Access.
- **S-GW:** Serving Gateway.
- **SIMO:** Single Input Multiple Output.
- **SINR:** Signal to Interference and Noise Ratio.
- **SISO:** Single Input Single Output.
- **SRS:** Sounding Reference Signal.
- **SSS:** Secondary Synchronization Signal.
- **TB:** Transport Block.
- **TDD:** Time Division Duplex.
- **TTI:** Transmission Time Interval.
- **UE:** User Equipment.
- **ULSCH:** Uplink Shared Channel.

Apéndice C

Oficio ARCOTEL



Oficio Nro. ARCOTEL-CZO6-2017-0260-OF

Cuenca, 07 de marzo de 2017

Asunto: REQUIEREN INFORMACION - MEDICIONES DE CALIDAD DE LAS OPERADORAS DEL SERVICIO MÓVIL AVANZADO EN TECNOLOGÍA LTE.

Señor
Josue Bernardo Vanegas Samaniego
En su Despacho

De mi consideración:

En atención a su oficio SN de 09 de febrero de 2017, ingresado a la Agencia de Regulación y Control de las Telecomunicaciones el día 09 de febrero de 2017 con número de trámite ARCOTEL-DEDA-2017-002451-E, mediante el cual se solicita facilitar las mediciones de calidad de servicio móvil avanzado en tecnología LTE realizadas en la ciudad Cuenca, informo que ésta Coordinación Zonal no dispone por el momento de las mediciones solicitadas.

Atentamente,

Documento firmado electrónicamente

Ing. Edgar Efraín Ochoa Figueroa
COORDINADOR ZONAL 6

Referencias:
- ARCOTEL-DEDA-2017-002451-E

Anexos:
- arcotel-deda-2017-002451-e_josue_vanegas_samaniego.pdf

Copia:
Señora
Lorena Del Carmen Vélez Arizaga
Servidor Público

ea/ap

Bibliografía

- [1] 3rd Generation Partnership Project (2016); Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation (Release 13).
- [2] ACADEMY, Q. W. (2014). LTE Release 8/9 Technical Overview e-Book. San Diego, EEUU .
- [3] BARRERA, D. (2015). Simulación y Análisis de cobertura para tecnología LTE en el centro histórico de la ciudad de Cuenca.. Cuenca, Ecuador.
- [4] BURILLO, V., ARIET, L., HERNANDO, J., AND VODAFONE, F. (2002) Sistemas de Comunicaciones Móviles de Tercera Generación IMT-2000 (UMTS), Fundación Airtel Vodafone. Madrid, España.
- [5] COX, C. (2012) An Intoduction to LTE, John Wiley and Sons. Gran Bretaña.
- [6] HERNANDO, J., MENDO, L., AND RIERA, J. (2010) Comunicaciones Móviles, Madrid, España.
- [7] HOLMA, H., AND TOSKALA, A. (2009).LTE for UMTS- OFDMA and SC-FDMA Based Radio Access. Inglaterra, Gran Bretaña: WILEY.
- [8] HOLMA, H., AND TOSKALA, A. (2010) WCDMA for UMTS: HSPA Evolution and LTE, John Wiley and Sons.
- [9] ITU. (2012) Especificaciones detalladas de las interfaces radioeléctricas terrenales de las comunicaciones móviles internacionales-avanzadas (IMT-Avanzadas), Ginebra, Suiza.
- [10] MADRIGAL, J., ALTAMIRANO, D., AND LEÓN, R. (Noviembre, 2015). Plataforma para la Enseñanza de la Capa Física de LTE R8 Usando Matlab. Revista Tecnológica ESPOL, VOL 28, 29-41. Sangolquí, Ecuador.



- [11] MOLINEROS, V. (2010). Modelamiento y simulación de sistemas MIMO. Guayaquil, Ecuador.
- [12] MORALES, D., SÁNCHEZ, J., AND GÓMEZ, G. (2007). Análisis de rendimiento de sistemas MIMO-SDM OFDM Málaga, España.
- [13] PEDRINI, L. (2012). What is Retransmission, ARQ and HARQ?. Fecha Consulta: Marzo 05, 2017, de TelecomHall Sitio web: <http://www.telecomhall.com/what-is-retransmission-arq-and-harq.aspx>.
- [14] SESIA, S., TOUFIK, I., AND BAKER, M. (2014) LTE- The UMTS Long Term Evolution, John Wiley and Sons, Gran Bretaña.
- [15] VALLEJO, D., AND ALVAREZ, R. (2012). Estudio y simulación de turbo códigos utilizando el algoritmo MAP y SOVA. Quito, Ecuador.
- [16] ZARRINKOUB, H. (2014) Understanding LTE with MATLAB, John Wiley and Sons.

UNIVERSIDAD DE CUENCA
desde 1867